Synchronous Bandwidth Allocation for Real-Time Communications with the Timed-Token MAC Protocol

Ching-Chih Han, Kang G. Shin, *Fellow*, *IEEE*, and Chao-Ju Hou, *Senior Member*, *IEEE*

Abstract—One of the key issues in tailoring the timed-token MAC protocol for real-time applications is synchronous bandwidth allocation (SBA), whose objective is to meet both the protocol and deadline constraints. The former constraint requires that the total time allocated to all nodes for transmitting synchronous messages should not exceed the target token rotation time. The latter constraint requires that the minimum time available for a node to transmit its synchronous messages before their deadlines should be no less than the maximum message transmission time. Several nonoptimal local SBA schemes and an optimal global SBA scheme have been proposed [1], [2], [3], [8], [17], [29]. Local SBA schemes use only information available locally to each node and are thus preferred to global schemes because of their lower network-management overhead. If optimal local SBA schemes, if any, can be devised, they will be superior to their global counterparts both in performance and in ease of network management. In this paper, we formally prove that there does not exist any optimal local SBA scheme. We also propose an optimal global SBA scheme which has an $O(nM)$ polynomial-time worst-case complexity, where n is the number of synchronous message streams in the system and M is the time complexity for solving a linear programming problem with $3n$ constraints and n variables.

Index Terms—Real-time communications, timed-token protocol, synchronous bandwidth allocation, FDDI.

1 Introduction

There has been an increasing need of timely, dependable communication services for such real-time systems as multimedia, automated factories, and industrial process controls. Such systems are usually realized by completing the execution of a number of cooperating/communicating tasks on multiple processors before their deadlines imposed by the corresponding mission/function. To meet the communication requirement, network architectures and protocols are required to provide the users with a convenient means of guaranteeing message-delivery delay bounds.

The problem of guaranteeing the timely delivery of messages has been studied by numerous researchers, especially in the context of voice/video data transmission over a data network and in the context of communications in embedded real-time systems. Their efforts have been directed mainly toward designing medium access control (MAC) protocols for multiaccess networks with time-constrained messages. In the case of token ring or token bus networks, Strosnider et al. [26] use a priority-based variation of the token passing protocol, called the *token scheduling* protocol, to implement the rate-monotonic scheduling algorithm [20] for network access control. Both IEEE 802.4 [4] (adopted for the Manufacturing Automation Protocol) [11] and FDDI [5] (developed by ANSI for high bandwidth fiber optic networks) use the *timed-token* MAC protocol.

There are two classes of messages in the timed-token protocol: *synchronous* and *asynchronous*. Synchronous messages are usually associated with delivery deadlines. Asynchronous messages have no such timing constraint. At network initialization, a protocol parameter, called *Target Token Rotation Time* (TTRT), is negotiated among the nodes to specify the expected token rotation time. Each node i is assigned a portion, say H_i, of TTRT as its *synchronous bandwidth*, which is the maximum time node i is permitted to transmit its synchronous messages every time it receives the token. The total synchronous bandwidth allocated should not exceed TTRT minus various protocol-dependent overheads. Whenever a node receives the token, it transmits its synchronous messages, if any, up to H_i units of time. The node can transmit its asynchronous messages only if the time interval between the previous and current token arrivals is less than TTRT, i.e., the token arrived *earlier* than expected. On the other hand, the assumption of a bounded token rotation time provides only a *necessary* condition for meeting message deadlines. If the synchronous bandwidth H_i allocated to node i is too small, then a message (with transmission time greater than H_i) may not be transmitted in time even if the token visits node i more than once after the message arrival. Consequently, in addition to the protocol constraint discussed above, H_i should be assigned so that the minimum time available for node i to transmit its synchronous messages after their

- C.-C. Han is with CreOsys Inc., 39560 Stevenson Pl. Suite 221, Fremont, CA 94539. E-mail: cchan@creosys.com.
- K.G. Shin is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2122. E-mail: kgshin@eecs.umich.edu.
- C.-J. Hou is with the Department of Electrical Engineering, Ohio State University, Columbus, OH 43210-1272. E-mail: jhou@ee.eng.ohio-state.edu.

Manuscript received 21 Apr. 1997; revised 24 Oct. 2000; accepted 10 Jan. 2001.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 104904.

arrivals but before their delivery deadlines is greater than or equal to the maximum total message transmission time during this period. This timing constraint in calculating H_is is called the *deadline constraint*.

Many researchers have studied the access time bounds and other timing properties of the timed-token protocol (e.g., the average and worst-case token rotation time), asynchronous message throughput, and the impact of tuning protocol parameters on the timing performance [10], [16], [22], [25], [28]. In particular, Johnson [16] and Sevcik and Johnson [25] proved that the *average* token cycle (rotation) time is bounded by TTRT and the *maximum* token cycle time is bounded by $2 \times \mathrm{TTRT}$. Agrawal et al. [2], [1], [3], [8] extended Johnson's result and proved that the time elapsed between k consecutive token visits to a node is bounded by $k \times \mathrm{TTRT}$. They also formulated a *synchronous bandwidth allocation* (SBA) problem and attempted to calculate the synchronous bandwidth H_i that should be allocated to node i, for all i, to meet the protocol constraint and to guarantee the timely delivery of all synchronous messages.

As discussed in [3], SBA schemes can be classified as local or global, depending on the type of information they use in calculating H_i. A local SBA scheme uses only parameters available locally to node i, i.e., the message deadline and the maximum message transmission time of a synchronous message stream at node i, in addition to the TTRT known to all nodes. In a global scheme, each node i uses the parameters of all the other nodes' synchronous message streams (as well as its own) to compute H_i. Any change in a node's message stream parameters may require the global scheme to adjust the synchronous bandwidths allocated to *all* nodes since all nodes use these parameters in calculating their synchronous bandwidths. As the global SBA schemes use global information to allocate synchronous bandwidths, they are naturally expected to achieve better performance. However, local schemes are preferable to global schemes from the perspective of network management.

To the best of our knowledge, there is only one optimal global SBA scheme [8] and several nonoptimal local SBA schemes [1], [2], [3], [29] reported in the open literature. By an "optimal" SBA scheme, we mean an SBA scheme that finds a feasible set of H_is subject to the protocol and deadline constraints whenever such a set exists. In this paper, we formally prove, using the technique of adversary argument, that there does not exist any optimal local SBA scheme. Hence, whether to choose a nonoptimal local scheme or an optimal global scheme depends on the trade-off between the ease of network management and the performance improvement.

The nonexistence of optimal local schemes also motivates us to devise an optimal global scheme of polynomial-time complexity. Note that the only currently known optimal global SBA scheme (proposed in [8]) uses an iterative approach to find the synchronous bandwidth allocation and may not terminate. Even though a traditional engineering approach can be applied to terminate the algorithm at a certain point, such as forcing the algorithm to terminate when the improvement for the solution is smaller than a certain threshold, the synchronous bandwidths found in their algorithm may still be unusable since some of them are not large enough to meet the deadline constraints of the real-time messages. One important issue is to determine if there exists any polynomial-time optimal global SBA scheme. In this paper, we answer this question positively by proposing an optimal SBA scheme which has an $O(nM)$ polynomial-time worst-case complexity, where n is the number of synchronous message streams in the system and M is the time complexity for solving a linear programming problem with $3n$ constraints and n variables.

The rest of the paper is organized as follows: In Section 2, we discuss the synchronous message model used for real-time applications and give a brief overview of the timed-token MAC protocol. In Section 3, we present several timing properties for the timed-token MAC protocol and discuss the timing requirements imposed by the message streams with delivery deadlines on the protocol. In Section 4, we formulate the SBA problem and then present proof of nonexistence of optimal local SBA schemes. In Section 5, we describe our polynomial-time global optimal SBA scheme. We conclude the paper with Section 6.

2 Message Model and MAC Protocol

In this section, we first discuss the synchronous message model suitable for real-time applications. To make the paper self-contained, we then review the timed-token MAC protocol used in FDDI token rings and some of its timing properties. A more detailed description of the timed-token protocol and FDDI token rings can be found in [5], [15], [23], [24].

2.1 Message Model

Let n be the number of nodes in the system. Without loss of generality, we assume that there is one synchronous message stream at each node. As was discussed in [3], a more general token ring network in which a node may have more than one synchronous message stream can be transformed into an equivalent network with more nodes in which each node has only one synchronous message stream. We adopt a message model similar to the (r, T)-*smooth* traffic model [12], [13], in which the synchronous message stream at node i can be described by a 2-tuple (C_i, D_i), where:

- C_i is the maximum total time needed to transmit the messages that arrive at node i in *any* time interval of length D_i (or simply called the *maximum message transmission time*) and
- D_i is the *relative* transmission (delivery) deadline for the messages at node i, i.e., if a message arrives at node i at time t, it must be *transmitted* by time $t + D_i$.

Note that the above message model, called the (C, D)-smooth model, can be easily implemented by the *leaky bucket* [27] traffic shaping mechanism. Moreover, this model is, in fact, a generalization of the real-time periodic message model adopted in [2], [8]. The reader is referred to [14] for a detailed discussion of these message models.

The "worst-case" scenario of the (C, D)-smooth model occurs when each synchronous message that arrives at node i has a message transmission time C_i. In such a case, the synchronous bandwidth allocated to node i after the arrival and before the deadline of each message must be at least C_i. Since the *exact* time when a message arrives at node i is not known a priori, in order to guarantee the timely delivery of each synchronous message of node i, an SBA scheme must set the parameters of the MAC protocol in such a way that "the *minimum* time available for node i to transmit its synchronous messages in *any* time interval of length D_i is at least C_i."

2.2 MAC Protocol

The key idea of the timed-token MAC protocol is to control the token rotation time. A protocol parameter called the *target token rotation time* (TTRT) is determined at network initialization and specifies the expected token rotation time. Each node i is assigned a portion, say H_i, of the TTRT, known as its *synchronous bandwidth*, which is the maximum time a node is permitted to transmit its synchronous messages every time it receives the token. The token is then forced by the protocol to circulate with sufficient speed so that all nodes receive their allocated fractions of bandwidth for transmitting synchronous messages. Specifically, each node has two timers and one counter:

- The *token rotation timer* (TRT) records the time elapsed since the last token visit (if the TRT has not yet expired). It is initialized to TTRT and counts down 1) until it reaches zero or 2) until the token is received and the time elapsed since its last visit is less than TTRT. In either case, TRT is reset to TTRT and continues to count down.
- The *token holding timer* (THT) records the amount of time by which the token has arrived early. This time can be used to transmit asynchronous messages. It is initialized to zero, is set to the value of TRT when the token arrives early, and counts down during the transmission of asynchronous messages.
- The *late counter* (LC) records the number of times its TRT has expired since the token's last visit to the node. It is initialized to zero, is incremented whenever TRT expires, and is reset to zero each time the node receives the token.

After the TTRT value is negotiated among the nodes during network initialization, each node initializes its timers and counter as follows:

$$\mathrm{TRT} \leftarrow \mathrm{TTRT};\ \ \mathrm{THT} \leftarrow 0;\ \ \mathrm{LC} \leftarrow 0.$$

TRT is enabled during all ring operations and always counts down until one of the following three events occurs:

E1. *TRT reaches zero:* The following steps are taken: 1) TRT $\leftarrow$ TTRT and TRT continues to count down, and 2) LC $\leftarrow$ LC + 1.

E2. *The token arrives early:* This is identified by LC $= 0$ at the time of token arrival. In this case, the following steps are taken: 1) THT $\leftarrow$ TRT and THT counts down only during the transmission of asynchronous messages, 2) TRT $\leftarrow$ TTRT and TRT continues to count down, 3) asynchronous messages, if any, are transmitted until THT expires or until all asynchronous messages are transmitted, whichever occurs first, and 4) synchronous messages are transmitted up to H_i units of time or until all synchronous messages are transmitted, whichever occurs first.

E3. *The token arrives late:* This is identified by LC $\neq 0$ at the time of token arrival. In this case, the following steps are taken: 1) LC $\leftarrow 0$, 2) TRT is *not* reset and continues to count down, and 3) only synchronous messages can be transmitted up to H_i units of time and no asynchronous messages can be transmitted.

3 Protocol Timing Properties and Real-Time Requirements

In this section, we first discuss several interesting timing properties associated with the MAC protocol described above. We then discuss the timing requirements imposed on the parameters of the MAC protocol by the messages with delivery deadlines.

To facilitate the discussion and the subsequent derivation, we introduce the following notation:

- T: the TTRT of an FDDI network.
- θ_i: the latency between node i and its upstream neighbor, where the upstream neighbor is node $i-1$ if $i > 1$, or node n if $i = 1$. θ_i includes medium propagation delay, token transmission time, station latency, and token capture delay [25].
- Θ: the *ring latency*, i.e., $\Theta = \sum_{i=1}^{n} \theta_i$.
- Ω: the various protocol-dependent overheads.
- τ: the portion of the synchronous bandwidth unavailable for transmitting synchronous messages, i.e., $\tau = \Theta + \Omega$.
- $\vec{H}$: vector $(H_1, H_2, \ldots, H_n)$, where H_i is the synchronous bandwidth allocated to node i.
- $g_{c,i}$ ($a_{c,i}$): the time spent on transmitting synchronous (asynchronous) traffic on the cth token visit to node i.
- $C_{c,i}$: the length of the time interval between the $(c-1)\mathrm{th}$ token departure and the cth token departure from node i. By using the *circular sum* operator [25], $C_{c,i}$ can be expressed as:

$$C_{c,i} = \sum_{j,k=c-1,i+1}^{c,i} (g_{j,k} + a_{j,k}) + \tau,$$

where the double-index circular sum operator is defined as:

$$\sum_{j,k=\ell,m}^{c,i} p_{j,k} = \begin{cases} \sum_{k=m}^{i} p_{c,k} = p_{c,m} + p_{c,m+1} + \cdots + p_{c,i} & \text{if } \ell = c \text{ and } m \le i, \\ \sum_{k=m}^{n} p_{\ell,k} + \sum_{j=\ell+1}^{c-1} \sum_{k=1}^{n} p_{j,k} + \sum_{k=1}^{i} p_{c,k} & \text{if } \ell < c, \end{cases}$$

for $1 \le m, i \le n$ and $(\ell, m) \le (c, i)$ (i.e., $\ell < c$ or $\ell = c$ and $m \le i$), and the single-index circular sum operator is defined as:

$$\sum_{i=j}^{k} p_i = \begin{cases} p_j + p_{j+1} + \cdots + p_k & \text{if } 1 \le j \le k \le n, \\ p_j + p_{j+1} + \cdots \\ +p_n + p_1 + p_2 + \cdots + p_k & \text{if } 1 \le k < j \le n. \end{cases}$$

- $d_i(\ell)$: the time instant when the token departs from node i the ℓth time, $1 \le b \le n$ and $\ell \ge 1$.
- X_i: the *minimum* time available for node i to transmit its synchronous messages in the time interval $(t, t + D_i]$, for all $t \ge 0$.
- f_g, f_l: the functions which represent the global and local SBA schemes, respectively. That is, a global SBA scheme is represented as $\vec{H} = f_g(\vec{C}, \vec{D}, T, \tau)$, where $\vec{C} = (C_1, C_2, \ldots, C_n)$ and $\vec{D} = (D_1, D_2, \ldots, D_n)$, and a local allocation scheme is represented as $H_i = f_l(C_i, D_i, T, \tau)$, for $i = 1, 2, \ldots n$.

Note that, by the protocol description, we have[1]

$$g_{c,i} \le H_i, \text{ and } a_{c,i} \le \max(0, T - C_{c,i-1}) \tag{3.1}$$

for $c \ge 1$ and $1 \le i \le n$.

3.1 Timing Properties of Timed-Token MAC Protocol

The protocol constraint on the allocation of synchronous bandwidth requires that:

$$\sum_{i=1}^{n} H_i \le T - \tau. \tag{3.2}$$

Violation of the protocol constraint will make the ring unstable and oscillate between "claiming" and "operational" [21].

Recall that $d_b(\ell)$ is the time instant when the token departs from node b the ℓth time. Let $\Delta_{b,i}(\ell, c)$ be the time difference between a reference time point $d_b(\ell)$ and the time instant when the token departs from node i the cth time *after* $d_b(\ell)$. That is,

$$\Delta_{b,i}(\ell, c) = \begin{cases} d_i(\ell + c - 1) - d_b(\ell) & \text{if } 1 \le b < i \le n, \\ d_i(\ell + c) - d_b(\ell) & \text{if } 1 \le i \le b \le n. \end{cases}$$

Under the protocol constraint, Theorem 1 gives a timing property of the timed-token MAC protocol [16], [25].

Theorem 1 (Johnson and Sevcik). *For the timed-token MAC protocol, the worst-case token rotation time—the time interval between the ℓth token departure and the $(\ell+1)$th token departure from node b—is bounded by $T + \sum_{j=1}^{n} H_j + \tau$, i.e.,*

$$\Delta_{b,b}(\ell, 1) = d_b(\ell + 1) - d_b(\ell) \le T + \sum_{j=1}^{n} H_j + \tau \le 2 \cdot T,$$

for any $1 \le b \le n$, and $\ell \ge 1$.

1. This is actually a relaxed version of the original protocol which allows asynchronous transmission without requiring "lateness" to be carried forward from cycle to cycle. Refer to [25] for details.

In the following lemma, we give a more general result which will be used in proving the Generalized Johnson and Sevcik Theorem (Theorem 2 and Corollary 1).

Lemma 1. *For the timed-token MAC protocol, we have*

$$\Delta_{b,i}(\ell, 1) \le T + \sum_{j=b+1}^{i} (H_j + \theta_j) + \Omega \le T + \sum_{j=b+1}^{i} H_j + \tau,$$

for any $1 \le b, i \le n$ and $\ell \ge 1$.

Using the above lemma and following similar derivation steps as in [25], we can obtain a more general result on the upper bound of the time difference between $d_b(\ell)$ and the time instant when the token leaves node i the cth time after time $d_b(\ell)$:

Theorem 2. *For the timed-token MAC protocol, we have*

$$\begin{aligned} \Delta_{b,i}(\ell, c) &\le c \cdot T + \sum_{j=b+1}^{i} (H_j + \theta_j) + \Omega \\ &\le c \cdot T + \sum_{j=b+1}^{i} H_j + \tau \le (c+1) \cdot T, \end{aligned} \tag{3.3}$$

for any $1 \le b, i \le n$, $\ell \ge 1$, and $c \ge 1$.

The proof of Theorem 2 is given in Appendix A.

Note that if we use $\alpha_i(\ell)$ to denote the time of the ℓth token arrival at node i, then, for $i < n$, $d_i(\ell) = \alpha_{i+1}(\ell)$ and, for $i = n$, $d_n(\ell) = \alpha_1(\ell + 1)$ (assuming that latency/overhead is ignored). Therefore, it is easy to see that results similar to the above lemma/theorems can be derived for token arrival times. If we set $b = i$ in (3.3), we obtain the following corollary (Fig. 1).

Corollary 1 (Generalized Johnson and Sevcik). *For the timed-token MAC protocol, the time elapsed between any $c + 1$ consecutive token visits to a node is bounded by $c \cdot T + \sum_{j=1}^{n} H_j + \tau \le (c + 1) \cdot T$.*

A result similar to Corollary 1 was obtained by Agrawal et al. [2], [3], [9] using a more complicated approach.

3.2 Deadline Constraint Imposed by Real-Time Messages

Every synchronous message must be transmitted before its delivery deadline. Hence, the minimum time, X_i, available for node i to transmit its synchronous messages in a time interval $(t, t + D_i]$ should be no less than the required maximum message transmission time, C_i. Using Corollary 1, Agrawal et al. [3], [8] derived the following lower bound for the time available for a node to transmit its synchronous messages within a given interval of length D_i.

Theorem 3. *Let D_i be the deadline constraint of the synchronous messages arrived at node i ($1 \le i \le n$). The minimum amount of time, X_i, available for node i to transmit its synchronous messages during a time interval $(t, t + D_i]$ of length D_i is given by:*

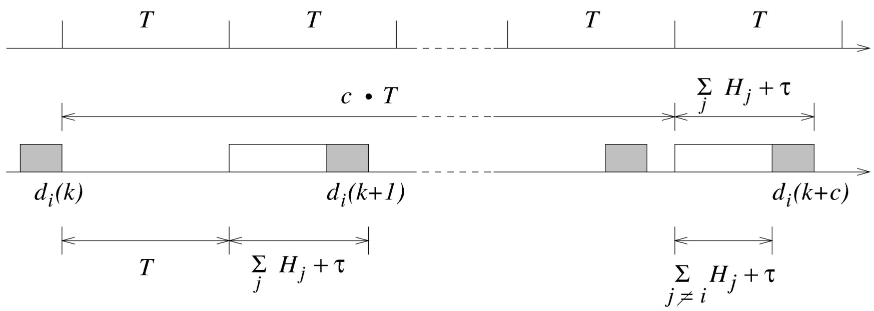

Fig. 1. Generalized Johnson and Sevcik Theorem.

$$X_i(\vec{H}) = (q_i - 1) \cdot H_i + \max\left(0, \min\left(r_i - \left(\sum_{j=1,\ldots,n,j\neq i} H_j + \tau\right), H_i\right)\right), \tag{3.4}$$

where $q_i = \lfloor D_i/T \rfloor$ *and* $r_i = D_i - q_i \cdot T$. *Any* D_i *may be represented as* $D_i = q_i \cdot T + r_i$.

Note that the time available for a node to transmit its synchronous messages within a time interval $(t, t + D_i]$ becomes *minimal* when t is the time of a token departure from the node. Also note that (3.4) can be rewritten as:

$$X_i(\vec{H}) = \begin{cases} (1)\ \ q_i \cdot H_i, & \text{if } r_i \geq \sum_j H_j + \tau, \\ (2)\ \ (q_i - 1) \cdot H_i + r_i \\ \quad - \sum_{j\neq i} H_j - \tau, & \text{if } \sum_{j\neq i} H_j + \tau < r_i < \sum_j H_j + \tau, \\ (3)\ \ (q_i - 1) \cdot H_i, & \text{if } r_i \leq \sum_{j\neq i} H_j + \tau. \end{cases} \tag{3.5}$$

Fig. 2 depicts the three cases. Since the time elapsed between any $(c+1)$ consecutive token visits is bounded by $c \cdot T + \sum_{j=1}^{n} H_j + \tau$:

Case 1. If $r_i \geq \sum_j H_j + \tau$, then $D_i \geq q_i \cdot T + \sum_j H_j + \tau$ and D_i can "accommodate" the q_ith token visit since time t.

Case 2. If $\sum_{j\neq i} H_j + \tau < r_i < \sum_j H_j + \tau$, then D_i can accommodate the first $(q_i - 1)$ token visits and part of the q_ith token visit (i.e., $r_i - (\sum_{j\neq i} H_j + \tau)$) since time t.

Case 3. If $r_i \leq \sum_{j\neq i} H_j + \tau$, then, in the worst case, $D_i = q_i \cdot T + r_i \leq q_i \cdot T + \sum_{j\neq i} H_j + \tau$ and D_i cannot accommodate the q_ith token visit since time t. However, $D_i \geq q_i \cdot T \geq (q_i - 1) \cdot T + \sum_j H_j + \tau$ and D_i can accommodate the first $(q_i - 1)$ token visits since time t.

For a message with transmission time C_i and deadline D_i that arrives at node i at time t, the timing requirement imposes the following deadline constraint:

$$X_i(\vec{H}) \geq C_i, \quad i = 1, 2, \ldots, n. \tag{3.6}$$

4 The SBA Problem and Nonexistence Proof

In this section, we first give a formal mathematical formulation of the SBA problem. We then give a proof of the nonexistence of optimal local SBA schemes of the form $H_i = f_l(C_i, D_i, T, \tau)$, for $i = 1, 2, \ldots, n$.

Problem 1 (The SBA Problem). An SBA scheme is an algorithm which, given the number of nodes (or synchronous message streams), n, the maximum message transmission time vector, $\vec{C} = (C_1, C_2, \ldots, C_n)$, the deadline vector, $\vec{D} = (D_1, D_2, \ldots, D_n)$, and the negotiated TTRT, T, allocates synchronous bandwidth, $\vec{H}$, to all the nodes subject to the following two constraints:

Protocol constraint: The computed SBA vector $\vec{H}$ must satisfy

$$\sum_{i=1}^{n} H_i \leq T - \tau. \tag{4.1}$$

Deadline constraint: The vector $\vec{H}$ must satisfy

$$X_i(\vec{H}) = (q_i - 1) \cdot H_i + \max\left(0, \min\left(r_i - \left(\sum_{j=1,\ldots,n,j\neq i} H_j + \tau\right), H_i\right)\right) \geq C_i, \tag{4.2}$$

where $q_i = \lfloor D_i/T \rfloor$, $r_i = D_i - q_i \cdot T = D_i - \lfloor D_i/T \rfloor T$, and $i = 1, 2, \ldots, n$.

For a global SBA scheme, this is accomplished through the function

$$\vec{H} = (H_1, H_2, \ldots, H_n) = f_g(\vec{C}, \vec{D}, T, \tau).$$

For a local SBA scheme, this is accomplished through the function $H_i = f_l(C_i, D_i, T, \tau)$, for $i = 1, 2, \ldots, n$.

A *feasible* solution for the SBA problem is a vector $\vec{H}$ that satisfies both the protocol and deadline constraints. An *optimal* global (local) SBA scheme is one that implements the function f_g (f_l) and finds a feasible solution whenever such a solution exists.

While an optimal global SBA scheme [8] and several nonoptimal local schemes [1], [2], [3], [29] have been proposed, it remains unknown if there exists any optimal local SBA scheme. The following theorem provides a formal proof of the nonexistence of optimal local SBA schemes.

Theorem 4. *There does not exist any optimal local SBA scheme for every fixed* $n \geq 3$, *where* n *is the number of nodes (synchronous message streams) in the system.*

Proof. It suffices for us to prove this theorem for $n = 3$ and $\tau = 0$. Thus, τ is dropped in the following discussion.

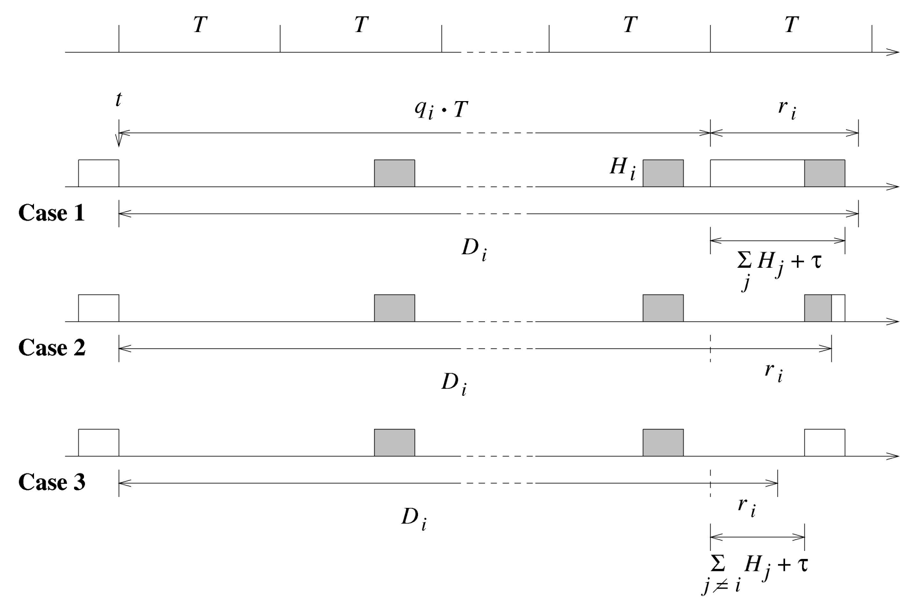

Fig. 2. Worst-case token visit scenarios in a time interval of length D_i.

For clarity of presentation, we outline the proof here and leave the detailed algebraic manipulation in Appendix B. Our proof is based on the technique of *adversary argument*, a detailed account of which can be found in [6]. Let L be any local SBA scheme and let A be the adversary. A first chooses (and fixes) the values for C_1, D_1, and T, and asks L for the value of H_1. Since L is a local SBA scheme, it should be able to give a value of H_1, say h, based only on the values of C_1, D_1, and T. After L gives A the value h of H_1, A chooses the values, C_i and D_i, for $i = 2, 3$, such that, with $H_1 = h$ given by L, it is impossible to find a feasible solution $\vec{H} = (h, H_2, H_3)$ for the SBA problem with the chosen T, C_i, D_i, for $i = 1, 2, 3$. However, a feasible solution does exist if H_1 is not restricted to be h. If, for every value h that A receives from L, A can always design an instance of the SBA problem such that the above situation occurs, then, by the adversary argument, we prove that L cannot be an optimal local SBA scheme since there are cases in which feasible solutions exist but L is not able to find one. Theorem 4 is thus proven. □

It is worth mentioning that, for $n = 2$, optimal local SBA schemes do exist. In fact, it can be shown that the local SBA scheme proposed in [29] is optimal for $n = 2$.

5 Polynomial-Time Optimal SBA Scheme

In the previous section, we have formally proven that there does not exist any optimal local SBA scheme (for every fixed $n \geq 3$). It remains unknown if there exists any polynomial-time optimal global SBA scheme. In this section, we first inspect the only known optimal global SBA scheme, MCA, proposed by Chen et al. [8], and show that MCA may not terminate. Then, we propose an optimal global SBA scheme.

5.1 Chen et al.'s Algorithm, MCA

For convenience of reference, we outline the optimal global SBA scheme, MCA [8], in Fig. 3 and discuss some of its properties below. Note that MCA assumes that $q_i \geq 2$, for all i. We first make the same assumption, and will discuss later how to relax this assumption.

Let Π be the set of $\vec{H}$s that satisfy the deadline constraint (4.1) (but not necessarily the protocol constraint (4.1)), i.e.,

$$\Pi = \{\vec{H} \mid X_i(\vec{H}) \geq C_i, \text{for all } i\}. \tag{5.1}$$

Also, for two given vectors $\vec{H}'$ and $\vec{H}''$, we define:

- $\vec{H}' = \vec{H}''$ if and only if $H'_i = H''_i$, for all i,
- $\vec{H}' \leq \vec{H}''$ if and only if $H'_i \leq H''_i$, for all i, and
- $\vec{H}' < \vec{H}''$ if and only if $\vec{H}' \leq \vec{H}''$ and $\vec{H}' \neq \vec{H}''$, i.e., $H'_i \leq H''_i$, for all i, and $H'_i < H''_i$, for some i.

Chen et al. proved the following theorem (Theorem 6.1 in [8]):

Theorem 5. *The set* Π *satisfies the following properties:*

P1. Π *is nonempty, i.e., (4.2) is solvable,*
P2. *There is a minimal element* $\vec{H}^*$ *in* Π, *i.e., for any* $\vec{H} \in \Pi$, $\vec{H}^* \leq \vec{H}$, *and*
P3. $\frac{C_i}{q_i} \leq H^*_i \leq \frac{C_i}{q_i - 1}$, *for all* i.

The proof of this theorem can be found in the technical report version of [8].

As shown in Fig. 3, MCA uses the procedure, called *Min_H*, to find the minimal element $\vec{H}^*$ in Π and then checks whether or not $\vec{H}^*$ satisfies the protocol constraint (4.1). If yes, $\vec{H}^*$ is a feasible solution to the input instance of the SBA problem. Otherwise, there is no feasible solution to the instance (since $\vec{H}^*$ is the minimal element in Π, if $\sum_{i=1}^{n} H^*_i > T - \tau$, then $\sum_{i=1}^{n} H_i > T - \tau$ for all $\vec{H} \in \Pi$). It is easy to see that if Procedure *Min_H* can always find $\vec{H}^*$, then MCA is an optimal SBA scheme (but not the converse). To find $\vec{H}^*$, Procedure *Min_H* uses an iterative approach,

Procedure *Min_H*;
1. For $i = 1$ to n do $H_i := \frac{C_i}{q_i}$;
2. Repeat
3. For $i = 1$ to n do calculate X_i as defined in Eq. (4.2);
4. For $i = 1$ to n do {
5. $b_i := C_i - X_i$;
6. If $b_i > 0$ then $H_i := H_i + \frac{b_i}{q_i - 1}$ };
7. Until none of b_is are larger than zero;

Allocation Scheme MCA
1. Call Procedure *Min_H* to obtain $\vec{H}^*$;
2. If $\sum_{i=1}^{n} H_i^* \leq T - \tau$ then return(success, $\vec{H}^*$)
3. else return(fail, nil)

Fig. 3. The optimal global SBA scheme, MCA, proposed by Chen et al. [8].

and first sets H_i to the minimum possible value of H_i^*, i.e., $\frac{C_i}{q_i}$, for all i (see **P3** in Theorem 5). If there exists any i such that H_i is not large enough to satisfy the deadline constraint (4.2) (i.e., $X_i < C_i$ for some i), then *Min_H* calculates the "deficiency" $b_i = C_i - X_i$ and increases H_i by the amount $\frac{b_i}{q_i - 1}$ for all i such that $b_i > 0$. The process repeats until all H_is are large enough to satisfy the deadline constraint.

Although Chen et al. proved that the value of $\vec{H}$ calculated in the Repeat-Until loop of Procedure *Min_H* is always less than or equal to $\vec{H}^*$ and will finally converge to $\vec{H}^*$, Procedure *Min_H* is *not* guaranteed to terminate. It is easy to find values of T, D_is (or q_is and r_is), C_is, and τ such that Procedure *Min_H* will never terminate. For example, let $T = 30$, $\tau = 0$, $q_i = 6$, $r_i = 24$, and $C_i = 30$, for $i = 1, 2, \ldots, 5$. Let $b_i^{(k)}$ denote the value of b_i at the kth iteration of the loop in their algorithm, then $b_i^{(k)} = (\frac{4}{5})^{k-1} > 0$, for all i. Even if we use a traditional engineering approach to terminate the algorithm at a certain point, such as forcing the algorithm to terminate when all b_is are smaller than a certain threshold, the values of H_is thus found are still unusable since some of them are not large enough to satisfy the deadline constraint (i.e., $b_i > 0$ for some i).

5.2 A New Polynomial-Time SBA Scheme

To remedy the deficiency that MCA may not terminate in polynomial time, we propose another algorithm to find the minimal element $\vec{H}^* \in \Pi$. The proposed algorithm is guaranteed not only to terminate but also to terminate in polynomial time. Before delving into the description of the algorithm, we first study the deadline constraint in more detail.

As discussed in Section 3, there are three possible cases for the value of $X_i(\vec{H})$ depending on which region r_i falls in (see (3.5) and Fig. 2). For ease of discussion, if $\vec{H}$ is unambiguous in the context, we say that H_i is in Region I, II, or III, if $r_i \geq \sum_j H_j + \tau$, $\sum_{j \neq i} H_j + \tau < r_i < \sum_j H_j + \tau$, or $r_i \leq \sum_{j \neq i} H_j + \tau$, respectively. It is easy to see that $X_i(\vec{H}^*) = C_i$, for all i, since if $X_i(\vec{H}^*) > C_i$ we can find another vector $\vec{H}'$ with $H_i' = H_i^* - \epsilon$ (where ϵ is a very small positive number) and $H_j' = H_j^*$ for $j \neq i$, which also satisfies the deadline constraint and, hence, contradicts that $\vec{H}^*$ is the minimal element in Π (this property will be used later). Therefore, we can conclude that if H_i^* is in Region I, II, or III, then H_i^* equals $\frac{C_i}{q_i}$, $\frac{C_i - (r_i - \sum_{j \neq i} H_j^* - \tau)}{q_i - 1}$, or $\frac{C_i}{q_i - 1}$, respectively. Moreover, if we know which region H_i^* falls in for each i, then the values of H_i^*s can be easily determined by solving the following system of n linear equations with n variables:

$$x_i = \begin{cases} \frac{C_i}{q_i}, & \text{if } H_i^* \text{ is in Region I}, \\ \frac{C_i - (r_i - \sum_{j \neq i} x_j - \tau)}{q_i - 1}, & \text{if } H_i^* \text{ is in Region II}, \\ \frac{C_i}{q_i - 1}, & \text{if } H_i^* \text{ is in Region III}, \end{cases} \tag{5.2}$$

for $i = 1, 2, \ldots, n$. Note that C_is, q_is, r_is, and τ are given numbers and x_is are the variables in the above system of linear equations. Since each H_i^* may fall in one of the three regions, if we try to find the vector $\vec{H}^*$ by guessing all the possibilities that H_i^*s may fall in, we need to solve the system of linear equations (5.2) 3^n times, and find the minimal $\vec{x}$ such that each of its elements indeed falls in the region we guessed. This implies that the SBA problem can be solved by an algorithm that is guaranteed to terminate in exponential time.

We propose an algorithm, Procedure *PT-Min_H*, which finds the minimal element $\vec{H}^*$ in Π in polynomial time. For ease of discussion, we call $\frac{C_i}{q_i}$, $\frac{C_i - (r_i - \sum_{j \neq i} H_j - \tau)}{q_i - 1}$, and $\frac{C_i}{q_i - 1}$ Formulas I, II, and III, respectively. Note that if H_i^* falls in Region I, II, or III, its correct value should be calculated according to Formula I, II, or III, respectively.

Procedure *PT-Min_H* (Fig. 4) works as follows: During the execution of PT-Min_H, F_i, $i = 1, 2, 3$, are the sets of indices of H_is whose current values are calculated using Formulas I, II, and III, respectively, and R_i, $i = 1, 2, 3$, are the sets of H_is whose current values fall in Regions I, II, and III, respectively. In Step 1, we first assume that H_i^* is in

Procedure PT-Min_H

Step 1. For $i = 1$ to n do $H_i := \frac{C_i}{q_i}$.
Let $F_1 := \{1, 2, \dots, n\}$, and $F_2 := F_3 := \emptyset$.

Step 2. Partition $\{1, 2, \dots, n\}$ into three subsets R_1, R_2, and R_3, where $R_1 = \{i \mid r_i \geq \sum_j H_j + \tau\}$, $R_2 = \{i \mid \sum_{j \neq i} H_j + \tau < r_i < \sum_j H_j + \tau\}$, and $R_3 = \{i \mid r_i \leq \sum_{j \neq i} H_j + \tau\}$.

Step 3. If $F_1 = R_1$ then $\vec{H}^*$ has been found and the algorithm terminates.

Step 4. Let $R := F_1 - R_1$, and $S := R \cup F_2$. Let $b_i := C_i - X_i(\vec{H})$, $a_i := q_i - 1$, and $\delta_i := \frac{C_i}{q_i - 1} - H_i$, for all $i \in S$. Solve the following linear programming (LP) formulation:

$$\text{maximize} \quad z = \sum_{i \in S} x_i \tag{5.3}$$

$$\text{subject to} \quad x_i \leq \frac{b_i}{a_i} + \frac{\sum_{j \in S, j \neq i} x_j}{a_i}, \tag{5.4}$$

$$x_i \leq \delta_i, \text{ and} \tag{5.5}$$

$$x_i \geq 0, \tag{5.6}$$

for all $i \in S$.
Let (x_i^*), $i \in S$, be the solution of the above LP.
Reset $H_i := H_i + x_i^*$, for all $i \in S$.
Reset $F_1 := R_1$, $F_2 := \{i \in S \mid x_i^* < \delta_i\}$, and $F_3 := F_3 \cup \{i \in S \mid x_i^* = \delta_i\}$.
Go to Step 2.

Fig. 4. Polynomial-time algorithm for finding the minimal element $\vec{H}^*$ in Π.

Region I and, hence, set $H_i := \frac{C_i}{q_i}$, for all i, set $F_1 := \{1, 2, \dots, n\}$, and $F_2 := F_3 := \emptyset$.

As the current values of H_is may not all fall in the regions as we expected, we find in Step 2 the correct region that each H_i falls into (according to the relationship between r_i and the current value of $\vec{H}$). If the formula used in calculating H_i matches the region that H_i really falls in, then the formula we used to calculate the value of H_i is correct. If this is true for all i, then all H_is have been calculated using the correct formulas and, hence, the algorithm terminates (Step 3). Otherwise, some of the H_is should have been calculated using Formula II or III, but were calculated using Formula I. As will be discussed later, only H_is with $i \in F_1$ may be calculated according to wrong formulas.

At the beginning of Step 4, $R = F_1 - R_1$ is the set of indices of H_is whose current values are calculated using Formula I but actually fall in Region II or III. For each $i \in R$, the deficiency, b_i, of X_i (i.e., the difference between the maximum message transmission time C_i and the minimum available transmission time $X_i(\vec{H})$) is larger than 0. This means that, to satisfy the deadline constraint, the synchronous bandwidth H_i of node i, for all $i \in R$, should be increased to compensate the deficiency. However, increasing H_is with $i \in R$ will further incur positive deficiency for all of the H_is with $i \in F_2$ and some of the H_is with $i \in R_1$. Since we are unaware of which H_is with $i \in R_1$ will incur positive deficiency, we can only take H_is with $i \in S = R \cup F_2$ into consideration and temporarily leave H_is with $i \in R_1$ fixed at the value $\frac{C_i}{q_i}$. Note that, since $\frac{C_i}{q_i - 1}$ is the maximum possible value of H_i^*, those H_is with $i \in F_3$ no longer need to be changed and, hence, are also fixed at $\frac{C_i}{q_i - 1}$. We then formulate and solve a linear programming (LP) problem to find the (maximum) values that H_is (*not* X_is) with $i \in S$ should be increased.

It is easy to see that, for each $i \in S = R \cup F_2$, the increase, x_i, in H_i makes the increase in $X_i(\vec{H})$ by $a_i \cdot x_i$ (note that $a_i = q_i - 1$). Therefore, H_i should be increased by (at least) $\frac{b_i}{a_i}$ to compensate for the deficiency b_i of X_i. However, due to the increase, x_j, of some other H_j with $j \in S$ and $j \neq i$, $X_i(\vec{H})$ will be decreased by the same amount x_j and, hence, the deficiency of X_i will be increased by x_j. If we let x_i be the amount of increase for H_i, for each $i \in S$, the actual deficiency of X_i will be $b_i + \sum_{j \in S, j \neq i} x_j$. Consequently, we expect to increase H_i by the amount $\frac{b_i}{a_i} + \frac{\sum_{j \in S, j \neq i} x_j}{a_i}$ ((5.4) in Step 4). To account for the other inequality (5.5), note that if $\frac{b_i}{a_i} + \frac{\sum_{j \in S, j \neq i} x_j}{a_i}$ equals $\delta_i = \frac{C_i}{q_i - 1} - H_i$, H_i will move from Region I or II to Region III and, hence, it need not be increased to a value larger than $\frac{C_i}{q_i - 1}$. That is, the amount of increase x_i for H_i never needs to be greater than $\min(\delta_i, \frac{b_i}{a_i} + \frac{\sum_{j \in S, j \neq i} x_j}{a_i})$. We then solve the LP with the $3 \cdot |S|$ constraints (including the nonnegativity constraints (5.6)) and the objective function (maximize) $z = \sum_{i \in S} x_i$, and increase H_i, for all $i \in S$, by the amount x_i^*, where (x_i^*) with $i \in S$ is the solution of the LP.

After the step $H_i := H_i + x_i^*$, for all $i \in S$, in Step 4, H_is with $i \in R_1$ are exactly those H_is whose values are fixed at

$\frac{C_i}{q_i}$ (Formula I) in the current iteration. Therefore, we reset $F_1 := R_1$. Similarly, H_is with $i \in S$ and $x_i^* < \delta_i$ are exactly those whose new values are calculated using Formula II and H_is with $i \in S$ and $x_i^* = \delta_i$ and with $i \in F_3$ are exactly those whose (new) values are calculated using Formula III or fixed at $\frac{C_i}{q_i - 1}$ in the current iteration. Therefore, we reset F_2 and F_3 accordingly. It is now clear that (after the reset of F_is) all H_is with $i \in F_2 \cup F_3$ are calculated using the correct formulas (with respect to the current $\vec{H}$). However, due to the increase of those H_js with $j \in S$, some H_is with $i \in F_1$ may now move from Region I to Region II or III. As a result, these H_is with $i \in F_1$ may still be calculated using the wrong formula and we need to go back to Step 2 to check if further changes are necessary.

We prove in Theorem 6 in Appendix C that the LP formulated in Step 4 has exactly one optimal solution (x_i^*) with $i \in S$ which satisfies either 1) $0 < x_i^* = \delta_i \leq \frac{b_i}{a_i} + \frac{\sum_{j \in S, j \neq i} x_j}{a_i}$ or 2) $0 < x_i^* = \frac{b_i}{a_i} + \frac{\sum_{j \in S, j \neq i} x_j}{a_i} < \delta_i$. Note that the above property of the unique optimal solution to the LP (i.e., either 1) or 2) is true for x_i^*) is the key point of the optimality of our algorithm since it implies that the new value of H_i (after being increased by x_i^*), for all $i \in S$, found in each iteration of Step 4 are not too large (i.e., the new value of $\vec{H}$ will be less than or equal to $\vec{H}^*$). With this property, if we can also show that the value of $\vec{H}$ found in each iteration of Step 4 will eventually converge to $\vec{H}^*$ in a polynomial number of iterations, then we prove the correctness (the optimality) and the polynomial-time complexity of Procedure PT-Min_H. The detailed proofs are given in Theorem 7 in Appendix C. In summary, we show in Theorem 7 that PT-Min_H can always find the minimal element $\vec{H}^*$ in Π in at most $O(nM)$ time, where n is the number of nodes (synchronous message streams) in the system and M is the time complexity for solving an LP with $3n$ constraints and n variables. Note that, although the famous simplex method for solving LP has an exponential-time worst-case complexity,[2] LP has been proven to be polynomial-time solvable [7], [18], [19]. Therefore, Procedure PT-Min_H is a polynomial-time algorithm.

In the above discussion, we assume that $q_i \geq 2$, for all i. As was mentioned earlier, with a little modification, the above SBA scheme can also handle the case with $q_i < 2$ for some i. If, for some i, $q_i = 0$, there does not exist any synchronous bandwidth allocation that guarantees the messages on node i can always meet their deadlines. If, for some i, $q_i = 1$, then, since $X_i(\vec{H}) = \max(0, \min(r_i - (\sum_{j \neq i} H_j + \tau), H_i)) \leq H_i$ for all $\vec{H}$, in order to satisfy the deadline constraint, we must have $H_i \geq C_i$. It is also easy to see that if $q_i = 1$, then H_i never needs to be larger than C_i since if there exists a feasible SBA $\vec{H}$ with $H_i > C_i$, then $\vec{H}'$ with $H_i' = C_i$ and $H_j' = H_j$ for $j \neq i$ is also a feasible SBA. Therefore, we can set $H_i = C_i$ for all $i \in \{j \mid q_j = 1\}$ and combine these H_is into the term τ, i.e., we can set $\tau' = \tau + \sum_{i \in \{j | q_j = 1\}} C_i$ and substitute τ' for τ in the protocol and deadline constraints. After we find the SBA $\vec{H}^*$ for the modified constraints, we must also check if the deadline constraint is satisfied for those X_is with $q_i = 1$.

2. However, in practice, the simplex method performs exceedingly well.

6 Conclusion

In this paper, we consider the synchronous bandwidth allocation (SBA) problem for the timed-token MAC protocol, formally prove the nonexistence of optimal local SBA schemes, and present an optimal global SBA scheme which is guaranteed to find a feasible solution in polynomial time for allocating synchronous bandwidth whenever such an allocation exists. The polynomial-time optimal global SBA algorithm described in this paper is, to the best of our knowledge, the first one in the literature. The existence of this algorithm also implies that the SBA problem is polynomial-time solvable (for global schemes).

The nonexistence proof of optimal local schemes and the proposed polynomial-time optimal global scheme suggest that the decision to use a nonoptimal local scheme or an optimal global scheme depends on the trade-off between the network management and the performance improvement. Another direction that is worthy of pursuit is to use the proposed optimal global scheme as a baseline scheme to study the performance of currently known nonoptimal local schemes. Note that, using the optimal global scheme, one can decide for each input instance of the SBA problem if feasible solutions exist. Based on the performance of these local schemes, we can then further characterize the trade-offs.

Appendix A
Proof of Theorem 2

Proof of Theorem 2. We prove the theorem for the case of $1 \leq i < b \leq n$. The proof for the other case is similar and thus omitted. Let $Q_{c,i} = c \cdot T + \sum_{j=b+1}^{i} (H_j + \theta_j) + \Omega$ and $R_{c,i} = \Delta_{b,i}(\ell, c) = d_i(\ell + c) - d_b(\ell)$ (for the case of $1 \leq i < b \leq n$) and let $G_{c,i} \triangleq Q_{c,i} - R_{c,i}$. We want to show that $G_{c,i} \geq 0$.

The proof is by contradiction. Assume that the xth token visit to node y after time $d_b(\ell)$ is the first visit after $d_b(\ell)$ for which $G_{c,i}$ is negative. Then, $G_{x,y} < 0$, but $G_{j,k} \geq 0$ for $1, b+1 \leq j, k < x, y$. First, x must be ≥ 2 because

$$G_{1,i} = \left(T + \sum_{j=b+1}^{i} (H_j + \theta_j) + \Omega \right) - \Delta_{b,i}(\ell, 1) \quad \geq 0,$$

where the inequality comes from Lemma 1. Now, we consider two cases:

Case 1: $g_{\ell+x,y} + a_{\ell+x,y} \leq H_y$. Consider the relationship between $G_{x,y}$ and $G_{x,y-1}$:

$$\begin{aligned}G_{x,y} - G_{x,y-1} &= (Q_{x,y} - Q_{x,y-1}) - (R_{x,y} - R_{x,y-1}) \\ &= (H_y + \theta_y) - (d_y(\ell + x) - d_{y-1}(\ell + x)) \\ &= H_y - (g_{\ell+x,y} + a_{\ell+x,y}) \quad \geq 0.\end{aligned}$$

Hence, $G_{x,y} \geq G_{x,y-1}$.

Case 2: $g_{\ell+x,y} + a_{\ell+x,y} > H_y$. Since $H_y \geq g_{\ell+x,y}$ (see (3.1)), we know that $a_{\ell+x,y} > 0$ in this case. That is, the $(\ell + x)$th token visit to node y occurs early or $C_{\ell+x,y-1} = \sum_{j,k=\ell+x-1,y}^{\ell+x,y-1}(g_{j,k} + a_{j,k}) + \tau < T$ and, hence, from (3.1), we have:

$$0 < a_{\ell+x,y} \leq \max(0, T - C_{\ell+x,y-1}) = T - C_{\ell+x,y-1}.$$

Consider the relationship between $G_{x,y}$ and $G_{x-1,y-1}$:

$$\begin{aligned}&G_{x,y} - G_{x-1,y-1} = (Q_{x,y} - Q_{x-1,y-1}) - (R_{x,y} - R_{x-1,y-1}) \\ &= (T + H_y + \theta_y) - (d_y(\ell + x) - d_{y-1}(\ell + x - 1)) \\ &= T + H_y - (C_{\ell+x,y-1} + g_{\ell+x,y} + a_{\ell+x,y}) \\ &= [(T - C_{\ell+x,y-1}) - a_{\ell+x,y}] + (H_y - g_{\ell+x,y}) \quad \geq 0.\end{aligned}$$

Hence, $G_{x,y} \geq G_{x-1,y-1}$.

We showed that $G_{x,y}$ was no less than $G_{x,y-1}$ in **Case 1** and no less than $G_{x-1,y-1}$ in **Case 2**, implying that x, y was not the first visit for which $G_{x,y}$ is negative. This contradiction shows that our assumption must be false and, thus, the theorem is proven. □

APPENDIX B
PROOF OF THEOREM 4

Proof of Theorem 4. We discuss how A chooses the instances for the SBA problem and jeopardizes the optimality claim of any local SBA scheme L. A first chooses:

$$C_1 = \frac{2}{3}T \text{ and } D_1 = 2\frac{2}{3}T,$$

where T is the TTRT and can be any fixed positive number. Suppose L computes a value h of H_1 based on the given values of C_1, D_1, and T. We consider three cases for the h value L computes: $h < \frac{1}{3}T$, $h = \frac{1}{3}T$, and $h > \frac{1}{3}T$.

C1. $h < \frac{1}{3}T$: For this case, we have

$$X_1(\vec{H}) \leq 2H_1 < \frac{2}{3}T = C_1,$$

which means that the deadline constraint for message stream 1 is violated. However, if A chooses $C_2 = C_3 = 0$,[3] it is easy to see that $(H_1, H_2, H_3) = (\frac{1}{3}T, 0, 0)$ is a feasible solution.

C2. $h = \frac{1}{3}T$: A can choose

$$C_2 = \frac{1}{3}T + \epsilon, C_3 = 0, \text{ and } D_2 = 2T,$$

where $0 < \epsilon \leq \frac{1}{6}T$. It is easy to see that, in order to satisfy the deadline constraint for message stream 2, H_2 must be larger than or equal to $\frac{1}{3}T + \epsilon$. Substituting $H_1 = h = \frac{1}{3}T$, $H_2 \geq \frac{1}{3}T + \epsilon$, and $H_3 = 0$ into $X_1(\vec{H})$, we have:

$$X_1(\vec{H}) = H_1 + \left(\frac{2}{3}T - H_2\right) \leq \frac{2}{3}T - \epsilon < C_1,$$

which means that the deadline constraint for message stream 1 is violated. However, $(H_1, H_2, H_3) = (\frac{1}{3}T + \epsilon, \frac{1}{3}T + \epsilon, 0)$ is a feasible solution for the chosen C_is, D_is, and T since $X_1(\vec{H}) = H_1 + (\frac{2}{3}T - H_2) = \frac{2}{3}T = C_1$, $X_2(\vec{H}) = H_2 = \frac{1}{3}T + \epsilon = C_2$, and $H_1 + H_2 = \frac{2}{3}T + 2\epsilon \leq T$, i.e., the deadline and protocol constraints are satisfied.

C3. $h > \frac{1}{3}T$: A can choose

$$C_2 = C_3 = \frac{1}{3}T \text{ and } D_2 = D_3 = D_1.$$

Since $C_2 = C_3$, $D_2 = D_3$, and L is a local scheme, L will compute $H_2 = H_3$. And, since

$$X_2(\vec{H}) = H_2 + \max\left(0, \min\left(\frac{2}{3}T - H_1 - H_3, H_2\right)\right),$$

we consider the following three subcases:

SC1. $X_2(\vec{H}) = 2H_2$, i.e.,

$$0 \leq H_2 \leq \frac{2}{3}T - H_1 - H_3:$$

Since $H_2 = H_3$ and $H_1 > \frac{1}{3}T$, we have

$$0 \leq H_2 \leq \frac{1}{3}T - \frac{1}{2}H_1 < \frac{1}{6}T.$$

But then, we have

$$X_2(\vec{H}) = 2H_2 < \frac{1}{3}T = C_2,$$

which means that the deadline constraint for message stream 2 is violated.

SC2: $X_2(\vec{H}) = H_2 + (\frac{2}{3}T - H_1 - H_3)$, i.e.,

$$0 \leq \frac{2}{3}T - H_1 - H_3 \leq H_2:$$

For this subcase, we have

$$\begin{aligned}X_2(\vec{H}) &= H_2 + \left(\frac{2}{3}T - H_1 - H_3\right) \\ &= \frac{2}{3}T - H_1 < \frac{1}{3}T = C_2,\end{aligned}$$

which means that the deadline constraint for message stream 2 is violated.

SC3: $X_2(\vec{H}) = H_2$, i.e., $\frac{2}{3}T - H_1 - H_3 \leq 0$: For this case, in order to satisfy the deadline constraint for message streams 2 and 3, we must have $H_2 = H_3 \geq C_2 = C_3 = \frac{1}{3}T$. But then, we get

$$H_1 + H_2 + H_3 > T,$$

which means that the protocol constraint is violated.

However, one can readily see that

3. Note that we can think that there is no message stream i in the system if $C_i = 0$.

$$H_1 = \frac{1}{3}T \text{ and } H_2 = H_3 = \frac{1}{6}T$$

is a feasible solution for the chosen C_is, D_is, and T since

$$X_1(\vec{H}) = H_1 + \left(\frac{2}{3}T - H_2 - H_3\right) = \frac{2}{3}T = C_1,$$

$$X_2(\vec{H}) = X_3(\vec{H}) = H_2 + \left(\frac{2}{3}T - H_1 - H_3\right)$$

$$= \frac{1}{3}T = C_2 = C_3,$$

and

$$H_1 + H_2 + H_3 = \frac{2}{3}T < T,$$

i.e., the deadline and protocol constraints are satisfied.

From **C1–C3**, Theorem 4 follows. □

Appendix C

Correctness and Time Complexity of PT-Min_H

To prove the correctness and derive the time complexity of Procedure PT-Min_H (Fig. 4), we need the following Theorem 6 and Lemmas 2 and 3.

Theorem 6 shows that there is a unique optimal solution to the LP formulated in Step 4 of Procedure PT-Min_H (Fig. 4), and this optimal solution satisfies some condition ((C.5)-(C.6)) which bounds the value of each element of the optimal solution.

Theorem 6. *Consider the following linear programming formulation:*

$$\text{maximize} \qquad z = \sum_{i=1}^{k} c_i \cdot x_i \tag{C.1}$$

$$\text{subject to} \qquad x_i \leq \frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j}{a_i}, \tag{C.2}$$

$$x_i \leq \delta_i \text{ and} \tag{C.3}$$

$$x_i \geq 0, \tag{C.4}$$

for $i = 1, 2, \ldots, k$. *If* $a_i > 0$, $b_i \geq 0$, $\delta_i > 0$, $c_i > 0$, *for all* i, *and there exists at least one index* j *such that* $b_j > 0$, *then:*

1. *The optimal value of the objective function* z *exists and is bounded,*
2. $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_k^*)$ *satisfies the following condition:*

$$\text{either} \qquad x_i^* = \delta_i \leq \frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j^*}{a_i} \tag{C.5}$$

$$\text{or} \qquad x_i^* = \frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j^*}{a_i} < \delta_i \tag{C.6}$$

for each $i = 1, 2, \ldots, k$, *where* $\mathbf{x}^*$ *is any optimal solution to the LP,*

3. *Any vector* $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ *satisfying (C.5)-(C.6) must have all positive components, i.e.,* $y_i > 0$, $\forall\, i$, *and*
4. *There is a* unique *vector satisfying (C.5)-(C.6), i.e.,* $\mathbf{x}^*$ *is the only vector satisfying (C.5)-(C.6) (and, hence,* $\mathbf{x}^*$ *is the unique optimal solution to the LP).*

Proof. The solution space of (C.2)-(C.4) is 1) nonempty since at least the zero vector $(0, 0, \ldots, 0)$ is a feasible solution and 2) bounded since any feasible solution $(x_1, x_2, \ldots, x_k)$ of (C.2)-(C.4) satisfies the constraint $0 \leq x_i \leq \delta_i$, for all i. Since the solution space of (C.2)-(C.4) is nonempty and bounded, the optimal value of the objective function z exists and is bounded and, hence, 1 is true.

Let $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_k^*)$ be a feasible solution of (C.2)-(C.4) which maximizes z (note that $x_i^* \geq 0$, for all i). Each x_i^*, $i = 1, 2, \ldots, k$, must satisfy (C.5)-(C.6) since if there exists an index l such that $x_l < \delta_l$ and $x_l < \frac{b_l}{a_l} + \frac{\sum_{j \neq l} x_l}{a_l}$, then $(x_1', x_2', \ldots, x_k')$ with $x_l' = x_l^* + \epsilon$ and $x_i' = x_i^*$, for all $i \neq l$, and ϵ sufficiently small is also a feasible solution for (C.2)-(C.4) and

$$\sum_{i=1}^{k} c_i \cdot x_i' = \sum_{i=1}^{k} c_i \cdot x_i^* + c_l \cdot \epsilon > \sum_{i=1}^{k} c_i \cdot x_i^*,$$

which contradicts the assumption that $(x_1^*, x_2^*, \ldots, x_k^*)$ maximizes z. Therefore, 2 is proven.

We next show that 3 is true. Let $(y_1, y_2, \ldots, y_k)$ be any vector satisfying (C.5)-(C.6). If there is an index i such that $y_i = 0$, then, since $\delta_i > 0$, we have $y_i = \frac{b_i}{a_i} + \frac{\sum_{j \neq i} y_i}{a_i} = 0$. Therefore, $b_i = 0$ and $\sum_{j \neq i} y_i = 0$, which implies $y_i = 0$, for all i. But this, in turn, implies that $b_i = 0$, for all i, which contradicts the assumption that there exists at least one index j such that $b_j > 0$.

Finally, we prove, by contradiction, that there is only one vector that satisfies (C.5)-(C.6). Assume that $\mathbf{x}' = (x_1', x_2', \ldots, x_k')$ and $\mathbf{x}'' = (x_1'', x_2'', \ldots, x_k'')$ are two distinct vectors that satisfy (C.5)-(C.6). Since $\mathbf{x}' \neq \mathbf{x}''$, they differ in at least one component. Without loss of generality, we can assume $x_l' < x_l''$. It is easy to see that $x_l' = \frac{b_l}{a_l} + \frac{\sum_{j \neq l} x_j'}{a_l} < \delta_l$ since $x_l'' \leq \delta_l$ and if $x_l' = \delta_l$, then $x_l'' \leq x_l'$, contradicting the assumption that $x_l' < x_l''$. Now, since

$$x_l' = \frac{b_l}{a_l} + \frac{\sum_{j \neq l} x_l'}{a_l} < x_l'' \leq \frac{b_l}{a_l} + \frac{\sum_{j \neq l} x_l''}{a_l},$$

we have $\sum_{j \neq l} x_j' < \sum_{j \neq l} x_j''$ and, hence,

$$\sum_j x_j' < \sum_j x_j''. \tag{C.7}$$

If there exists an h such that $x_h' > x_h''$, by the same argument as above, we have $\sum_{j \neq h} x_j' > \sum_{j \neq h} x_j''$ and $\sum_j x_j' > \sum_j x_j''$, which contradicts (C.7). Therefore,

$$x'_i \leq x''_i, \text{ for all } i. \tag{C.8}$$

Moreover, we claim that:

1. If $x'_i = x''_i$, then $x'_i = x''_i = \delta_i > 0$, and
2. If $x'_i < x''_i$, then $0 < x'_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x'_j}{a_i} < \delta_i$ and either
$$0 < x''_i = \delta_i \leq \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x''_j}{a_i}$$
or
$$0 < x''_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x''_j}{a_i} < \delta_i.$$

Note that if $x'_i = x''_i < \delta_i$, then $x'_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x'_j}{a_i} = x''_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x''_j}{a_i}$ and, hence, $\sum_{j\neq i} x'_j = \sum_{j\neq i} x''_j$. But this, in turn, implies that $\sum_j x'_j = \sum_j x''_j$, which, again, contradicts (C.7). Conversely, if $x'_i = \delta_i$, then, since $x'_i \leq x''_i \leq \delta_i$, we have $x''_i = \delta_i = x'_i$. Therefore, $x'_i = x''_i$ if and only if $x'_i = \delta_i$, and $x'_i < x''_i$ if and only if $x'_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x'_j}{a_i}$. Moreover, from 2, we know that $x'_i > 0$ and $x''_i > 0$, for all i. This concludes the proof of the above claim.

Now, let

$$U = \left\{ i \mid x'_i = \delta_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x'_j}{a_i} \right\},$$
$$V = \{ i \mid x'_i = x''_i = \delta_i \} - U,$$

and

$$W = \{ i \mid x'_i < x''_i \} \cup U.$$

(Note that $U = V \cup W$ and $V \cap W = \emptyset$.) Since, for all $i \in V$, $0 < x'_i = \delta_i < \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x'_j}{a_i}$ and, for all $i \in W$, $0 < x'_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x'_j}{a_i} \leq \delta_i$, $(x'_1, x'_2, \ldots, x'_k)$ is a solution for the following system of k linear equations:

$$x_i = \delta_i, \text{ for } i \in V, \tag{C.9}$$

and

$$x_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x_j}{a_i}, \text{ for } i \in W. \tag{C.10}$$

Now, there are two cases to consider:

Case 1: The system of linear equations (C.9)-(C.10) has an infinite number of feasible solutions.

Since the solution space of a system of linear equations with an infinite number of solutions is connected, given a solution of the system we must be able to find another solution of the system such that their corresponding components are very close to each other. And, since $x'_i = \delta_i < \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x'_j}{a_i}$, for $i \in V$, and $0 < x'_i < \delta_i$, for $i \in W$, there must exist another solution $(y'_1, y'_2, \ldots, y'_k)$ for the system of linear equations (C.9)-(C.10) such that $y'_i = x'_i = \delta_i < \frac{b_i}{a_i} + \frac{\sum_{j\neq i} y'_i}{a_i}$, for all $i \in V$, and $0 \leq y'_i < \delta_i$, for all $i \in W$. Therefore, $(y'_1, y'_2, \ldots, y'_k)$ is also a solution for (C.5)-(C.6).

Now, from (C.7) and (C.8), we have, for all i, either $x'_i \leq y'_i$ or $y'_i \leq x'_i$ and there exists an index j such that $x'_j \neq y'_j$. Without loss of generality, assume $y'_i \leq x'_i$, for all i. Since both $(x'_1, x'_2, \ldots, x'_k)$ and $(y'_1, y'_2, \ldots, y'_k)$ are solutions for the system of linear equations (C.9)-(C.10), we have that $(y_1, y_2, \ldots, y_k)$ with $y_i = y'_i - t \cdot (x'_i - y'_i)$, for any real number t, is also a feasible solution for (C.9)-(C.10). Now, if we gradually increase t (starting from 0), eventually we will reach a situation where either $y_l = 0$, for some $l \in W$ or $y_l = \delta_l = \frac{b_l}{a_l} + \frac{\sum_{j\neq l} y_j}{a_l}$, for some $l \in V$. For the former situation, we have $y_i = \delta_i < \frac{b_i}{a_i} + \frac{\sum_{j\neq i} y_j}{a_i}$, for all $i \in V$, and $y_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} y_j}{a_i} \leq \delta_i$, for all $i \in W$, and $y_l = 0$. Therefore, $(y_1, y_2, \ldots, y_k)$ satisfies (C.5)-(C.6), but $y_l = 0$, which contradicts part 3 of the theorem stating that $y_i > 0$, for all i. For the latter situation, we just substitute x'_i for the role of x''_i, and substitute y_i for the role of x'_i in the whole proof, and repeat the proof. Eventually, we will get to a point as the previous situation or as the situation to be discussed next (Case 2).

Case 2: $(x'_1, x'_2, \ldots, x'_k)$ is the only solution for the system of linear equations (C.9)-(C.10),.

For this case, we consider the following system of linear equations:

$$x_i = 0, \text{ for } i \in V, \tag{C.11}$$

and

$$x_i = \frac{\bar{b}_i}{a_i} + \frac{\sum_{j\neq i} x_j}{a_i}, \text{ for } i \in W, \tag{C.12}$$

where, for $i \in W$,

$$\bar{b}_i = b_i + \sum_{j\neq i} x''_j - a_i \cdot \delta_i, \text{ if } x''_i = \delta_i < \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x''_j}{a_i} \tag{C.13}$$

and

$$\bar{b}_i = 0, \text{ otherwise (i.e., if } x''_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x''_j}{a_i} \leq \delta_i). \tag{C.14}$$

Note that, since the system of linear equations (C.9)-(C.10), has a unique solution, it cannot be true that $x''_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x''_i}{a_i} < \delta_i$, for all $i \in W$, since, otherwise, $(x''_1, x''_2, \ldots, x''_k)$ is also a solution for the system of linear equations (C.9)-(C.10). This means that $\bar{b}_i \geq 0$, for all $i \in W$, and there is at least one i such that $\bar{b}_i = b_i + \sum_{j\neq i} x''_i - a_i \cdot \delta_i > 0$. Since the system of linear equations (C.9)-(C.10) has a unique solution, so does the system of linear equations (C.11)-(C.12) (note that both linear systems have the same constraint matrix A, and $det(A) \neq 0$.) Let $(y'_1, y'_2, \ldots, y'_k)$ be the unique solution of the system of linear equations (C.11)-(C.12). By a similar argument as we prove part 3 of the theorem, we know that $y'_i > 0$, for all $i \in W$. Now, let us consider $(y_1, y_2, \ldots, y_k)$ with $y_i = x''_i + y'_i$. For each $i \in V$, $y'_i = 0$ and $y_i = x''_i = \delta_i$. For each $i \in W$, if $x''_i = \delta_i < \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x''_i}{a_i}$, we have

$$\begin{aligned}
y_i &= x''_i + y'_i = x''_i + \frac{\bar{b}_i}{a_i} + \frac{\sum_{j\neq i} x''_j}{a_i} \\
&= \delta_i + \left(\frac{b_i}{a_i} + \frac{\sum_{j\neq i} x''_j}{a_i} - \delta_i + \frac{\sum_{j\neq i} y'_j}{a_i}\right) = \frac{b_i}{a_i} + \frac{\sum_{j\neq i}(x''_j + y'_j)}{a_i} \\
&= \frac{b_i}{a_i} + \frac{\sum_{j\neq i} y_j}{a_i};
\end{aligned}$$

otherwise (i.e., $x''_i = \frac{b_i}{a_i} + \frac{\sum_{j\neq i} x''_j}{a_i} \leq \delta_i$), we have

$$\begin{aligned}
y_i &= x''_i + y'_i = x''_i + \frac{\bar{b}_i}{a_i} + \frac{\sum_{j\neq i} x''_j}{a_i} \\
&= \left(\frac{b_i}{a_i} + \frac{\sum_{j\neq i} x''_j}{a_i}\right) + \frac{\sum_{j\neq i} y'_j}{a_i} = \frac{b_i}{a_i} + \frac{\sum_{j\neq i}(x''_j + y'_j)}{a_i} \\
&= \frac{b_i}{a_i} + \frac{\sum_{j\neq i} y_j}{a_i}.
\end{aligned}$$

Therefore, $(y_1, y_2, \ldots, y_k)$ is also a solution for the system of linear equations (C.9)-(C.10). But, since $y_i = x''_i + y'_i > x''_i \geq x'_i$, for $i \in W$, it means that the system of linear equations (C.9)-(C.10) has more than one solution, which contradicts the assumption of this case.

From Cases 1 and 2, 4 is proven, i.e., there is only one solution, $(x^*_1, x^*_2, \ldots, x^*_k)$, satisfying (C.5)_(C.6), which is also the unique optimal solution to the LP. □

To facilitate further discussion, we define the following notation: Let F_i^- and F_i^+ denote the values of F_i before and after the execution of Step 4 in an iteration of the loop from Step 2 to Step 4, respectively, and let $\vec{H}^-$ and $\vec{H}^+$ be similarly defined. Let $V \in \{R_i\text{s}, R, S, b_i\text{s}, \delta_i\text{s}\}$ denote the value of the corresponding variable in the current iteration of Step 4[4] and let V^+ denote the value of the corresponding variable in the next iteration of Step 4, i.e.,

$$\begin{aligned}
R_1^+ &= \{i \mid r_i \geq \sum_j H_j^+ + \tau\}, \\
R_2^+ &= \{i \mid \sum_{j\neq i} H_j^+ + \tau < r_i < \sum_j H_j^+ + \tau\}, \\
R_3^+ &= \{i \mid r_i \leq \sum_{j\neq i} H_j^+ + \tau\}, \\
R^+ &= F_1^+ - R_1^+, \\
S^+ &= R^+ \cup F_2^+, \\
b_i^+ &= C_i - X_i(\vec{H}^+),
\end{aligned}$$

and

$$\delta_i^+ = \frac{C_i}{q_i - 1} - H_i^+.$$

As mentioned in Section 5, F_1, F_2, and F_3 are the sets of indices of H_is whose current values are calculated using Formulas I, II, and III, respectively. Suppose H_i is calculated by Formula I (i.e., $H_i = \frac{C_i}{q_i}$). If the delivery deadline of the messages at node i is changed to $D_i^\# = q_i^\# \cdot T + r_i^\# = (q_i + 1) \cdot T$, then the deadline constraint for the messages at node i is guaranteed to be satisfied since, then, $X_i^\#(\vec{H}) = (q_i^\# - 1) \cdot H_i = C_i$. A similar statement also holds for H_is that are calculated according to Formula III. Thus, for $\# \in \{-, +\}$, we define

- $q_i^\# = q_i + 1$, $r_i^\# = 0$, for $i \in F_1^\#$,
- $q_i^\# = q_i$, $r_i^\# = r_i$, for $i \in F_2^\#$,
- $q_i^\# = q_i$, $r_i^\# = 0$, for $i \in F_3^\#$,
- $$\begin{aligned} X_i^\#(\vec{H}) &= (q_i^\# - 1) \cdot H_i \\ &+ \max\left(0, \min\left(r_i^\# - \left(\sum_{j=1,\ldots,n, j\neq i} H_j + \tau\right), H_i\right)\right), \end{aligned}$$

  and
- $\Pi^\# = \{\vec{H} \mid \bar{X}_i^\#(\vec{H}) \geq C_i, \ \text{for all } i\}$.

We will show that $\vec{H}^-$ ($\vec{H}^+$) is the minimal element in Π^- (Π^+).

Since, in Step 4, we fix the H_is with $i \in R_1$ and $i \in F_3^-$ at the values calculated by Formulas I and III, respectively, we also define:

- $q'_i = q_i + 1$, $r'_i = 0$, for $i \in R_1 = F_1^- - R$,
- $q'_i = q_i$, $r'_i = r_i$, for $i \in S = R \cup F_2^- = (F_1^- - R_1) \cup F_2^-$,
- $q'_i = q_i$, $r'_i = 0$, for $i \in F_3^-$,
- $$\begin{aligned} X'_i(\vec{H}) &= (q'_i - 1) \cdot H_i \\ &+ \max\left(0, \min\left(r'_i - \left(\sum_{j=1,\ldots,n, j\neq i} H_j + \tau\right), H_i\right)\right), \end{aligned}$$

  and
- $\Pi' = \{\vec{H} \mid X'_i(\vec{H}) \geq C_i, \ \text{for all } i\}$.

The reason we define the above notation is to show some relationships among $\vec{H}'$ (which is the minimal element in Π'), $\vec{H}^-$, $\vec{H}^+$, and $\vec{H}^*$. Their relationships are important in proving the correctness of Procedure PT-Min_H.

4. Note that these variables do not change in Step 4 after they are initially set at the beginning of Step 4.

We will use the following assumptions in Lemmas 2 and 3 and Theorem 7. (We will show that these assumptions are the invariants of the loop from Step 2 to Step 4).

A1. $\vec{H}^- \leq \vec{H}^*$,

A2. $H_i^- = \frac{C_i}{q_i}$, for $i \in F_1^-$,

$$\frac{C_i}{q_i} < H_i^- = \frac{C_i - (r_i - \sum_{j \neq i} H_j^* - \tau)}{q_i - 1} < \frac{C_i}{q_i - 1},$$

for $i \in F_2^-$, and $H_i^- = \frac{C_i}{q_i - 1}$, for $i \in F_3^-$,

A3. $b_i = 0$, for all $i \in F_2^-$, $b_i > 0$, for all $i \in R = F_1^- - R_1$, and $\delta_i > 0$, for all $i \in S$, and

A4. $\vec{H}^-$ is the minimal element in Π^-.

Assumption **A1** says that the value, $\vec{H}^-$, found for $\vec{H}$ before each iteration of Step 4 is always less than or equal to $\vec{H}^*$. Note that $\vec{H}^-$ is the minimal element in Π^- (Assumption **A4**), but not necessarily equal to the minimal element $\vec{H}^*$ in Π.

The following lemma proves some properties of the minimal element $\vec{H}'$ in Π'. These properties will be used in the proof of Lemma 3. In Lemma 3, we will show that $\vec{H}^+ = \vec{H}'$.

Lemma 2. *If **A1-A4** are true, then the minimal element $\vec{H}'$ in Π' satisfies $H_i' = H_i^-$, for $i \notin S$, and $H_i^- \leq H_i' \leq H_i^*$ and $X_i(\vec{H}') = C_i$, for all $i \in S$.*[5]

Proof. Let $H_i'' = H_i^*$, for $i \in S$, and $H_i'' = H_i^-$ $(\leq H_i^*)$, for $i \notin S$. We have

$$\text{for } i \in R_1,\ X_i'(\vec{H}'') = q_i \cdot H_i'' = q_i \cdot H_i^- = X_i^-(\vec{H}^-) = C_i,$$

$$\text{for } i \in F_3^-,\ X_i'(\vec{H}'') = (q_i - 1) \cdot H_i'' = (q_i - 1) \cdot H_i^- = X_i^-(\vec{H}^-) = C_i, \text{ and}$$

$$\begin{aligned}
&\text{for } i \in S,\ X_i'(\vec{H}'') = (q_i - 1) \cdot H_i'' + \max\left(0, \min\left(r_i - \left(\sum_{j \neq i} H_j'' + \tau\right), H_i''\right)\right) \\
&= (q_i - 1) \cdot H_i^* + \max\left(0, \min\left(r_i - \left(\sum_{j \neq i} H_j'' + \tau\right), H_i^*\right)\right) \\
&\geq (q_i - 1) \cdot H_i^* + \max\left(0, \min\left(r_i - \left(\sum_{j \neq i} H_j^* + \tau\right), H_i^*\right)\right) \\
&= X_i(\vec{H}^*) = C_i.
\end{aligned}$$

Therefore, $\vec{H}'' \in \Pi'$. Since $\vec{H}'' \leq H_i^*$ and $\vec{H}'$ is the minimal element in Π', we have $\vec{H}' \leq \vec{H}'' \leq \vec{H}^*$. Since $X_i' = X_i, \forall i \in S$, and $X_i'(\vec{H}') = C_i$, we have $X_i(\vec{H}') = C_i$, for all $i \in S$. Since $r_i' = r_i^- = 0$, $q_i' = q_i^-$, and $X_i'(\vec{H}') = X_i^-(\vec{H}^-) = C_i$, for $i \notin S$, it is easy to see that $H_i' = H_i^- = \frac{C_i}{q_i}$, for $i \in R_1$, and $H_i' = H_i^- = \frac{C_i}{q_i - 1}$, for $i \in F_3^-$.

Now, we prove that $H_i' \geq H_i^-$, for all $i \in S$. Note that:

1. $X_i^-(\vec{H}) = q_i \cdot H_i$, for $i \in F_1^-$ and, hence, for $i \in R = F_1^- - R_1$,
2. $$\begin{aligned} X_i^-(\vec{H}) &= X_i'(\vec{H}) \\ &= (q_i - 1) \cdot H_i + \max\left(0, \min\left(r_i - \sum_{j \neq i} H_j - \tau, H_i\right)\right), \end{aligned}$$ for $i \in F_2^-$, and
3. $X_i^-(\vec{H}) = (q_i - 1) \cdot H_i$, for $i \in F_3^-$.

Let $H_i'' = \min(H_i', H_i^-)$ (hence, $H_i'' \leq H_i^-$). Since, for $i \notin S$, $H_i' = H_i^-$, we have $H_i'' = H_i^-$. Also, since, for $i \in F_1$, $H_i^- = \frac{C_i}{q_i}$, we have $H_i' \geq H_i^-$, and hence, $H_i'' = H_i^-$, for $i \in F_1$. Therefore, if $H_i' < H_i^-$, then we must have $i \in F_2^-$. Now, if there exists an index $i \in F_2^-$ such that $H_i' < H_i^-$, then $H_i'' < H_i^-$ and $\vec{H}'' < \vec{H}^-$. Moreover, since

$$\text{for } i \in F_1^-,\ X_i^-(\vec{H}'') = q_i \cdot H_i'' = q_i \cdot H_i^- = X_i^-(\vec{H}^-) = C_i,$$

$$\text{for } i \in F_3^-,\ X_i^-(\vec{H}'') = (q_i - 1) \cdot H_i'' = (q_i - 1) \cdot H_i^- = C_i,$$

$$\begin{aligned}
&\text{for } i \in F_2^- \text{ and } H_i' \geq H_i^-,\ X_i^-(\vec{H}'') \\
&= (q_i - 1) \cdot H_i^- + \max\left(0, \min\left(r_i - \sum_{j \neq i} H_j'' - \tau, H_i^-\right)\right) \\
&= (q_i - 1) \cdot H_i^- \\
&\quad + \max\left(0, \min\left(r_i - \sum_{j \neq i} \min\left(H_j', H_j^-\right) - \tau, H_i^-\right)\right) \\
&\geq (q_i - 1) \cdot H_i^- + \max\left(0, \min\left(r_i - \sum_{j \neq i} H_j^- - \tau, H_i^-\right)\right) \\
&= X_i^-(\vec{H}^-) = C_i, \text{ and}
\end{aligned}$$

$$\begin{aligned}
&\text{for } i \in F_2^- \text{ and } H_i' < H_i^-,\ X_i^-(\vec{H}'') \\
&= (q_i - 1) \cdot H_i' + \max\left(0, \min\left(r_i - \sum_{j \neq i} H_j'' - \tau, H_i'\right)\right) \\
&= (q_i - 1) \cdot H_i' \\
&\quad + \max\left(0, \min\left(r_i - \sum_{j \neq i} \min\left(H_j', H_j^-\right) + \tau, H_i'\right)\right) \\
&\geq (q_i - 1) \cdot H_i' + \max\left(0, \min\left(r_i - \sum_{j \neq i} H_j' - \tau, H_i'\right)\right) \\
&= X_i'(\vec{H}') = C_i.
\end{aligned}$$

Therefore, $\vec{H}'' \in \Pi^-$ and, by Theorem 5, there exists a vector $\vec{H}'''$ with $\vec{H}''' \leq \vec{H}''$ $(< \vec{H}^-)$ such that $\vec{H}'''$ is the minimal element in Π^-, which contradicts the assumption that $\vec{H}^-$ is the minimal element in Π^-. Therefore, there cannot exist any index i such that $H_i' < H_i^-$. □

The following lemma proves that Assumptions **A1-A4** are loop invariants.

Lemma 3. *If Assumptions **A1-A4** are true and assume R is nonempty,*[6] *then*

1. $\vec{H}^- < \vec{H}^+ \leq \vec{H}^*$,
2. $H_i^+ = \frac{C_i}{q_i}$, *for* $i \in F_1^+$,

$$\frac{C_i}{q_i} < H_i^+ = \frac{C_i - (r_i - \sum_{j \neq i} H_j^* - \tau)}{q_i - 1} < \frac{C_i}{q_i - 1},$$

for $i \in F_2^+$, *and* $H_i^+ = \frac{C_i}{q_i - 1}$, *for* $i \in F_3^+$,

5. Actually, we will show in Lemma 3 that $H_i^- < H_i'$, for all $i \in S$.

6. Note that Step 4 will be executed only when R is nonempty.

3. $b_i^+ = 0$, *for all* $i \in F_2^+$, *and* $b_i^+ > 0$, *for all* $i \in R^+ = F_1^+ - R_1^+$, *and* $\delta_i^+ > 0$, *for all* $i \in S^+$, *and* $R_1^+ = \{i \mid r_i \geq \sum_j H_j^+ + \tau\}$, *and*
4. $\vec{H}^+$ *is the minimal element in* Π^+.

Proof. We first show that $\vec{H}^+$ is the minimal element in Π'. From the algorithm, we know that, for each $i \notin S$, $H_i^+ = H_i^- \leq H_i^*$. For each $i \in S$, $H_i^+ = H_i^- + x_i^*$, where (x_i^*) with $i \in S$ is the optimal solution of the LP found in Step 4. From **A3**, we have, in the LP in Step 4, $a_i > 0$, $b_i \geq 0$, $\delta_i > 0$, for all $i \in S$, and $b_i > 0$, for all $i \in R$. Also, since $R \neq \emptyset$, by Theorem 6, we know that (x_i^*) with $i \in S$ is the unique solution such that each x_i^*, $i \in S$, satisfies (C.5)-(C.6) in Theorem 6 and, moreover, $x_i^* > 0$, for all $i \in S$. For convenience of discussion, we define $x_i^* = 0$, for all $i \notin S$, therefore, $H_i^+ = H_i^- + x_i^*$, for all $i = 1, 2, \ldots, n$, and $\sum_{j \in S, j \neq i} x_j^* = \sum_{j \neq i} x_j^*$.

By Lemma 2, the minimal element $\vec{H}'$ in Π' is such that $H_i' = H_i^- \leq H_i^*$, for $i \notin S$, and $H_i^- \leq H_i' \leq H_i^*$ and $X_i(\vec{H}') = C_i$, for all $i \in S$. Let $x_i = H_i' - H_i^-$ (note that $x_i = 0$, for all $i \notin S$, and $x_i \geq 0$, for all $i \in S$). We will next show that each $x_i, i \in S$, satisfies (C.5)-(C.6) in Theorem 6. And, since (x_i^*) with $i \in S$ is the unique solution such that each x_i^*, $i \in S$, satisfies (C.5)-(C.6), we can conclude that $x_i = x_i^*$ and, hence, $\vec{H}^+ = \vec{H}'$.

Since $S = (F_1^- - R_1) \cup F_2^-$, for all $i \in S$, we have $r_i < \sum_j H_j^- + \tau$ and

$$X_i(\vec{H}) = (q_i - 1) \cdot H_i + \max\left(0, \min\left(r_i - \sum_{j \neq i} H_j^- + \tau, H_i^-\right)\right) = (q_i - 1) \cdot H_i + \max\left(0, r_i - \sum_{j \neq i} H_j^- - \tau\right)$$

and there are three cases to consider (for $i \in S$):

Case 1: H_i' falls in Region I, i.e., $r_i \geq \sum_j H_j' + \tau$.
In this case, we have

$$X_i(\vec{H}') = C_i$$
$$q_i \cdot H_i' = C_i$$
$$q_i \cdot (H_i^- + x_i) = C_i$$
$$x_i = \frac{C_i}{q_i} - H_i^-.$$

Since $x_i \geq 0$, we have $H_i^- \leq \frac{C_i}{q_i}$ and, since $H_i^- \geq \frac{C_i}{q_i}$, we have $H_i^- = \frac{C_i}{q_i}$, which implies that $i \in S \cap F_1^- = F_1^- - R_1$, and, hence, $i \notin R_1$. For $i \notin R_1$, we have

$$r_i < \sum_j H_j^- + \tau$$
$$r_i < \sum_j (H_j' - x_j) + \tau$$
$$\sum_j x_j < -(r_i - \sum_j H_j' - \tau) \leq 0.$$

But, this contradicts the fact that $H_i' \geq H_i^-$ ($x_i \geq 0$), for all $i \in S$. Therefore, this case cannot happen.

Case 2: H_i' falls in Region II, i.e.,

$$\sum_{j \neq i} H_j' + \tau < r_i < \sum_j H_j' + \tau.$$

In this case, we have

$$\sum_{j \neq i} H_j' + \tau < r_i$$
$$\sum_{j \neq i} H_j^- + \tau + \sum_{j \neq i} x_j < r_i$$
$$\sum_{j \neq i} x_j < r_i - \sum_{j \neq i} H_j^- - \tau < \max\left(0, r_i - \sum_{j \neq i} H_j^- - \tau\right)$$
$$a_i \cdot H_i^- + \sum_{j \neq i} x_j < X_i(\vec{H}^-) = C_i - b_i$$
$$b_i + \sum_{j \neq i} x_j < C_i - a_i \cdot H_i^-$$
$$\frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j}{a_i} < \delta_i = \frac{C_i}{a_i} - H_i^-,$$

and

$$X_i(\vec{H}') = C_i$$
$$a_i \cdot H_i' + \left(r_i - \sum_{j \neq i} H_j' - \tau\right) = C_i$$
$$a_i \cdot (H_i^- + x_i) + \left(r_i - \sum_{j \neq i} H_j^- - \tau - \sum_{j \neq i} x_j\right) = C_i$$
$$\left(a_i \cdot H_i^- + r_i - \sum_{j \neq i} H_j^- - \tau\right) + a_i \cdot x_i - \sum_{j \neq i} x_j\Bigg) = C_i.$$

If $r_i - \sum_{j \neq i} H_j^- - \tau < 0$, then $X_i(\vec{H}^-) = a_i \cdot H_i^-$, $b_i = C_i - X_i(\vec{H}^-) = C_i - a_i \cdot H_i^- = a_i \cdot \delta_i$. Therefore, $\delta_i = \frac{b_i}{a_i}$. But, we have shown that $\frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j}{a_i} < \delta_i$, which contradicts the fact that $\sum_{j \neq i} x_j \geq 0$. Therefore, we must have $r_i - \sum_{j \neq i} H_j^- - \tau \geq 0$ and, hence,

$$X_i(\vec{H}^-) = a_i \cdot H_i^- + \max\left(0, r_i - \sum_{j \neq i} H_j^- - \tau\right) = a_i \cdot H_i^- + r_i - \sum_{j \neq i} H_j^- - \tau$$

and

$$X_i(\vec{H}^-) + a_i \cdot x_i - \sum_{j \neq i} x_j = C_i$$
$$x_i = \frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j}{a_i}.$$

Therefore, for this case, we can conclude that

$$x_i = \frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j}{a_i} < \delta_i. \tag{C.15}$$

Case 3: H_i' falls in Region III, i.e., $r_i \leq \sum_{j \neq i} H_j' + \tau$.
In this case, we have

$$X_i(\vec{H}') = C_i$$
$$a_i \cdot H'_i = C_i$$
$$a_i \cdot (H_i^- + x_i) = C_i$$
$$x_i = \frac{C_i}{a_i} - H_i^- = \delta_i,$$

and, since $r_i \leq \sum_{j \neq i} H'_j + \tau$, we have

$$r_i \leq \sum_{j \neq i} H_j^- + \tau + \sum_{j \neq i} x_j$$
$$r_i - \sum_{j \neq i} H_j^- - \tau \leq \sum_{j \neq i} x_j.$$

If $r_i - \sum_{j \neq i} H_j^- - \tau \geq 0$, then

$$X_i(\vec{H}^-) - a_i \cdot H_i^- \leq \sum_{j \neq i} x_j$$
$$(C_i - b_i) - (C_i - a_i \cdot \delta_i) \leq \sum_{j \neq i} x_j$$
$$x_i = \delta_i \leq \frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j}{a_i}.$$

If $r_i - \sum_{j \neq i} H_j^- - \tau < 0$, then $X_i(\vec{H}^-) = a_i \cdot H_i^-$,

$$b_i = C_i - X_i(\vec{H}^-) = C_i - a_i \cdot H_i^- = a_i \cdot \delta_i.$$

And, since $\sum_{j \neq i} x_i \geq 0$, we have $x_i = \delta_i \leq \frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j}{a_i}$.

Therefore, for this case, we can conclude that

$$x_i = \delta_i \leq \frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j}{a_i}. \tag{C.16}$$

From Cases 1-3, we can conclude that either $0 \leq x_i = \frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j}{a_i} < \delta_i$ or $0 \leq x_i = \delta_i \leq \frac{b_i}{a_i} + \frac{\sum_{j \neq i} x_j}{a_i}$, for all $i \in S$. From Theorem 6, we know that there is a unique solution satisfying the above constraint and this solution is the solution (x_i^*) with $i \in S$ found in Step 4. Therefore, we have $H'_i - H_i^- = x_i^*$, for all $i \in S$. Moreover, since $x_i^* > 0$, for all $i \in S$, we have $H_i^- < H_i^+ = H'_i \leq H_i^*$, for all $i \in S$ (note that $H'_i = H_i^+ = H_i^-$, for all $i \notin S$), i.e., $\vec{H}^- < \vec{H}' = \vec{H}^+ \leq \vec{H}^*$. Therefore, 1 is proven.

Since $F_1^+ = R_1 = F_1^- - R$, we have $H_i^+ = H_i^- = \frac{C_i}{q_i}$, for $i \in F_1^+$. Similarly, with some mathematical manipulation, it is easy to check that $\frac{C_i}{q_i} < H_i^+ = \frac{C_i - (r_i - \sum_{j \neq i} H_j^* - \tau)}{q_i - 1} < \frac{C_i}{q_i - 1}$, for $i \in F_2^+$, and $H_i^+ = \frac{C_i}{q_i - 1}$, for $i \in F_3^+$. Hence, 2 can be easily proven.

Since $F_2^+ \subseteq S$, and $X_i(\vec{H}') = X_i(\vec{H}^+) = C_i$, for all $i \in S$, we have $b_i^+ = 0$. Since $R^+ = F_1^+ - R_1^+$, for all $i \in R^+$, H_i^+ is calculated using Formula I, but it actually falls in Region II, we have $b_i^+ > 0$. $S^+ = (F_1^+ - R_1^+) \cup F_2^+$ and, thus, for each $i \in S^+$, H_i^+ is calculated by Formula I or II and, hence, $\delta_i^+ > 0$. Therefore, 3 is proven.

Finally, we prove that 4 is true. Note that, for all $i \notin S$ and all $i \in F_2^+$, $X'_i = X_i^+$, and for all

$$i \in F_3^+ - F_3^- = \{j \in S \mid x_j^* = \delta_j\},$$

$H_i^+ = H'_i = \frac{C_i}{q_i - 1}$. Since $q_i^+ = q_i$ and $r_i^+ = 0$, for all $i \in F_3^+ \supseteq F_3^+ - F_3^-$, the minimal element $\vec{H}''$ in Π^+ must have $H''_i = H_i^+ = \frac{C_i}{q_i - 1}$, for $i \in F_3^+ - F_3^-$. If $\vec{H}^+ = \vec{H}'$ is not the minimal element $\vec{H}''$ in Π^+, then let $H'''_i = \min(H'_i, H''_i)$, for all i. By a similar calculation as we did in the proof of Lemma 2, we can show that $X'_i(\vec{H}''') = X_i^+(\vec{H}''') \geq C_i$, for all $i \notin S$ and all $i \in F_2^+$, and $X'_i(\vec{H}''') = C_i$, for all $i \in F_3^+ - F_3^-$. Therefore, $\vec{H}''' < \vec{H}'$ and $\vec{H}''' \in \Pi'$, which contradicts that $\vec{H}'$ is the minimal element in Π'. Therefore, $\vec{H}' = \vec{H}^+$ must be the minimal element in Π^+. □

We now prove the correctness of Procedure PT-Min_H and give its time complexity in the following theorem:

Theorem 7. *Let $\vec{H}^{(k)}$ be the value of $\vec{H}$ before the execution of the kth iteration of Step 3 in the loop from Step 2 to Step 4. We have:*

1. *Assumptions **A1-A4** are true for all iterations of the loop from Step 2 to Step 4; in particular, $\vec{H}^{(k)} \leq \vec{H}^*$, for all k,*
2. *Procedure PT-Min_H terminates after at most n iterations of Step 4 and, at termination (assuming after the lth iteration of Step 3), $\vec{H}^{(l)} = \vec{H}^*$, and*
3. *The time complexity of Procedure PT-Min_H is at most $O(nM)$, where M is the time complexity for solving an LP with $3n$ constraints and n variables.*

Proof. We prove 1 by induction on k. The first time when the algorithm goes to Step 3, $H_i^{(1)} = \frac{C_i}{q_i} \leq H_i^*$, for all i, $F_1 = \{1, 2, \ldots, n\}$, and $F_2 = F_3 = \emptyset$. It is easy to check that Assumptions **A1**, **A2**, and **A4** are true (note that Assumption **A3** is meaningful only if the algorithm goes to Step 4). If the algorithm terminates at the first iteration of Step 3, then, for all i, $r_i \geq \sum_j H_j^{(1)} + \tau$ and $X_i(\vec{H}^{(1)}) = C_i$. Therefore, $\vec{H}^{(1)} = \vec{H}^*$. If the algorithm goes to Step 4, it means that $R = F_1 - R_1 \neq \emptyset$ and $r_i < \sum_j H_j^{(1)} + \tau$, for $i \in R$. Therefore, $b_i = C_i - X_i(\vec{H}^{(1)}) > 0$ (note that $F_2 = \emptyset$). Hence, Assumption **A3** is true. Now, assume 1 is true at the beginning of the kth iteration of Step 4. By Lemma 2, 1 is still true after the execution of Step 4. Therefore, by the logic of induction proof, 1 is true.

Next, since, for each iteration, the size of F_1 will be decreased by $|R| = |F_1 - R_1|$ and, if $|R| = 0$, the algorithm will terminate. Therefore, the algorithm is guaranteed to terminate after (at most) n iterations of Step 4 and, when the algorithm terminates at the lth iteration of Step 3, $R = \emptyset$, which means that the current value of H_i is calculated by Formula I, II, or III (i.e., $i \in F_i$, $i = 1, 2, 3,$) if and only if H_i is in Region I, II, or III (i.e., $i \in R_i$, $i = 1, 2, 3$), respectively. Therefore, $X_i(\vec{H}^{(l)}) = C_i$, for all i (i.e., $\vec{H}^{(l)} \in \Pi$). Now, from 1 (i.e., $\vec{H}^{(l)} \leq \vec{H}^*$), we conclude that $\vec{H}^{(l)} = \vec{H}^*$. Thus, 2 is proven.

It is easy to see that the time complexity of the algorithm is dominated by the LP described in Step 4, which has at most $3n$ constraints and n variables. Since there are at most n iterations of Step 4, we need to solve the LP at most n times. Therefore, the time complexity of the algorithm is given as in 3. □

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments on an earlier version of this paper. The work in this paper was supported in part by the US Office of Naval Research under Grants N00014-92-J-1080 and N00014-94-1-0229, by the US National Science Foundation under Grants MIP-9203895 and ANIR-9625064. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] G. Agrawal, B. Chen, and W. Zhao, "Local Synchronous Capacity Allocation Schemes for Guaranteeing Message Deadlines with the Timed-Token Protocol," *Proc. Infocom,* pp. 186-193, Apr. 1993.
[2] G. Agrawal, B. Chen, W. Zhao, and S. Davari, "Guaranteeing Synchronous Message Deadlines with the Timed Token Protocol," *Proc. IEEE Int'l Conf. Distributed Computing Systems,* pp. 468-475, June 1992.
[3] G. Agrawal, B. Chen, W. Zhao, and S. Davari, "Guaranteeing Synchronous Message Deadlines with the Timed Token Medium Access Control Protocol" *IEEE Trans. Computers,* vol. 43, no. 3, pp. 327-350, Mar. 1994.
[4] "Token Passing Bus Access Method,"ANSI/IEEE Standard 802. 4-1985, 1985.
[5] "Fiber Distributed Data Interface (FDDI)—Token Ring Media Access Control (MAC),"ANSI X3.139, 1987.
[6] S. Baase, *Computer Algorithms: Introduction to Design and Analysis,* second ed. Addison-Wesley, 1991.
[7] M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali, *Linear Programming and Network Flows,* second ed. John Wiley & Sons, 1990.
[8] B. Chen, G. Agrawal, and W. Zhao, "Optimal Synchronous Capacity Allocation for Hard Real-Time Communications with the Timed Token Protocol," *Proc. 13th Real-Time Systems Symp.,* pp. 198-207, Dec. 1992.
[9] B. Chen and W. Zhao, "Properties of the Timed Token Protocol," Technical Report 92-038, Dept. of Computer Science, Texas A&M Univ., Oct. 1992.
[10] D. Dykeman and W. Bux, "Analysis and Tuning of the FDDI Media Access Control Protocol," *IEEE J. Selected Areas in Comm.,* vol. 6, no. 6, pp. 997-1010, July 1988.
[11] "Manufacturing Automation Protocol,"General Motors Corp., version 3.0, implementation release, May 1987.
[12] S.J. Golestani, "Congestion-Free Communication in High-Speed Packet Networks," *IEEE Trans. Comm.,* vol. 39, no. 12, pp. 1802-1812, Dec. 1991.
[13] S.J. Golestani., "A Framing Strategy for Congestion Management," *IEEE J. Selected Areas in Comm.,* vol. 9, no. 7, pp. 1065-1077, Sept. 1991.
[14] C.-C. Han, C.-J. Hou, and K.G. Shin, "On Slot Allocation for Time-Constrained Messages in Dual-Bus Networks," *IEEE Trans. Computers,* vol. 46, no. 7, pp. 756-767, July 1997.
[15] R. Jain, "FDDI: Current Issues and Future Plans," *IEEE Comm. Magazine,* pp. 98-105, Sept. 1993.
[16] M.J. Johnson, "Proof that Timing Requirements of the FDDI Token Ring Protocol Are Satisfied," *IEEE Trans. Comm.,* vol. 35, no. 6, pp. 620-625, June 1987.
[17] S. Kamat, N. Malcolm, and W. Zhao, "Performance Evaluation of a Bandwidth Allocation Scheme for Guaranteeing Synchronous Messages with Arbitrary Deadlines in an FDDI Network," *Proc. 14th Real-Time Systems Symp.,* pp. 34-43, Dec. 1993.
[18] L.G. Khachian, "A Polynomial Algorithm in Linear Programming," *Soviet Math. Doklady,* vol. 20, pp. 191-194, 1979.
[19] L.G. Khachian, "Polynomial Algorithms in Linear Programming," *USSR Computational Math. and Math. Physics,* vol. 20, pp. 53-72, 1980.
[20] C.L. Liu and J.W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment," *J. ACM,* vol. 20, no. 1, pp. 46-61, 1973.
[21] N. Malcolm and W. Zhao, "The Timed-Token Protocol for Real-Time Communications," *Computer,* vol. 27, no. 1, pp. 35-40, Jan. 1994.
[22] J.W.M. Pang and F.A. Tobagi, "Throughput Analysis of a Timer Controlled Token Passing Protocol under Heavy Load," *IEEE Trans. Comm.,* vol. 37, no. 7, pp. 694-702, July 1989.
[23] F.E. Ross, "FDDI—A Tutorial," *Computer,* vol. 24, no. 5, pp. 10-17, May 1986.
[24] F.E. Ross, "An Overview of FDDI: The Fiber Distributed Data Interface," *IEEE J. Selected Areas in Comm.,* vol. 7, no. 7, pp. 1043-1051, Sept. 1989.
[25] K.C. Sevcik and M.J. Johnson, "Cycle Time Properties of the FDDI Token Ring Protocol," *IEEE Trans. Software Eng.,* vol. 13, no. 3, pp. 376-385, Mar. 1987.
[26] J.K. Strosnider, T. Marchok, and J.P. Lehoczky, "Advanced Real-Time Scheduling Using the IEEE 802.5 Token Ring," *Proc. Ninth Real-Time Systems Symp.,* pp. 42-52, Dec. 1988.
[27] J.S. Turner, "New Directions in Communications (or Which Way to the Information Age)," *IEEE Comm. Magazine,* vol. 24, no. 10, pp. 8-15, Oct. 1986.
[28] A. Valenzano, P. Montuschi, and L. Ciminiera, "Some Properties of Timed Token Medium Access Protocols," *IEEE Trans. Software Eng.,* vol. 16, no. 8, pp. 858-869, Aug. 1990.
[29] Q. Zheng and K.G. Shin, "Synchronous Bandwidth Allocation in FDDI Networks," *Proc. ACM Multimedia '93,* pp. 31-38, Aug. 1993.

Ching-Chih Han received the BSE degree in electrical engineering in 1984 from National Taiwan University and the PhD degree in computer science in 1992 from the University of Illinois at Urbana-Champaign. Prior to joining Creosys, Inc., in Fremont, California, as a chief technology officer and a cofounder in February 1999, he was an associate professor in the Department of Applied Mathematics at National Sun Yat-sen University, Taiwan, from 1992 to 1994, an associate research scientist at the University of Michigan, Ann Arbor, from 1994 to 1996, an assistant professor in the Department of Electrical Engineering at Ohio State University from 1996 to 1997, and a senior software engineer at Broadvision, Inc., Redwood City, California, from 1997 to 1999. His areas of interest include Internet applications, real-time communication, multimedia applications, and real-time computing/scheduling.

Kang G. Shin received the BS degree in electronics engineering from Seoul National University, Seoul, Korea, in 1970, and both the MS and PhD degrees in electrical engineering from Cornell University, Ithaca, New York, in 1976 and 1978, respectively. He is a professor and founding director of the Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, the University of Michigan, Ann Arbor, Michigan. He has supervised the completion of 41 PhD theses and authored/coauthored more than 600 technical papers and numerous book chapters in the areas of distributed real-time computing and control, computer networking, fault-tolerant computing, and intelligent manufacturing. He has coauthored (jointly with C.M. Krishna) a textbook, *Real-Time Systems* (McGraw-Hill, 1997). In 1987, he received the Outstanding IEEE Transactions on Automatic Control Paper Award and the Research Excellence Award in 1989, Outstanding Achievement Award in 1999, and Service Excellence Award in 2000 from The University of Michigan. In 1985, he founded the Real-Time Computing Laboratory, where he and his colleagues are investigating various issues related to real-time and fault-tolerant computing.

His current research focuses on Quality of Service (QoS) sensitive computing and networking with emphases on timeliness and dependability. He has also been applying the basic research results to telecommunication and multimedia systems, embedded systems, and manufacturing systems.

From 1978 to 1982, he was on the faculty of Rensselaer Polytechnic Institute, Troy, New York. He has held visiting positions at the US Airforce Flight Dynamics Laboratory, AT&T Bell Laboratories, the Computer Science Division within the Department of Electrical Engineering and Computer Science at the University of California at Berkeley, and the International Computer Science Institute, Berkeley, California, the IBM T.J. Watson Research Center, and the Software Engineering Institute at Carnegie Mellon University. He also chaired the Computer Science and Engineering Division, EECS Department, the University of Michigan for three years beginning in January 1991.

He is an IEEE fellow and a member of the Korean Academy of Engineering, was the general chair of the 2000 IEEE Real-Time Technlogy and Applications Symposium, the program chair of the 1986 IEEE Real-Time Systems Symposium (RTSS), the general chair of the 1987 RTSS, the guest editor of the 1987 August special issue of *IEEE Transactions on Computers* on real-time systems, a program cochair for the 1992 International Conference on Parallel Processing, and served on numerous technical program committees. He also chaired the IEEE Technical Committee on Real-Time Systems during 1991-1993, was a Distinguished Visitor of the Computer Society of the IEEE, an editor of *IEEE Transactions on Parallel and Distributed Computing*, and an area editor of the *International Journal of Time-Critical Computing Systems* and *Computer Networks*.

Jennifer C. Hou received the BSE degree in electrical engineering from National Taiwan University in 1987, the MSE degree in electrical engineering and computer science (EECS) in 1989, the MSE degree in industrial and operations engineering in 1991, and the PhD degree in electrical engineering and computer science in 1993, all from the University of Michigan, Ann Arbor. From August 1993 to August 1996, she was an assistant professor in the Department of Electrical and Computer Engineering at the University of Wisconsin-Madison. Since August 1996, she has been with the Department of Electrical Engineering at the Ohio State University-Columbus, where she is currently an associate professor. She is the recipient of a US National Science Foundation CAREER award, Wisconsin/Hilldale Undergraduate/Faculty Research Fellowships, and Women in Science Initiative Awards in Wisconsin. Her current research interests are in the areas of network QoS, multicast routing, network modeling and simulation, resource management for distributed systems, and performance modeling/evaluation. She is a senior member of the IEEE and a member of the ACM.

▹ **IEEE Computer Society publications cited in this article can be found in our Digital Library at** http://computer.org/publications/dlib.